A HYBRID METAHEURISTIC ALGORITHM FOR THE CAPACITATED LOCATION-ROUTING PROBLEM

John Willmer Escobar
Department of Electronics, Computer Sciences and Systems (DEIS), University of Bologna
Viale Risorgimento 2, 40136 Bologna, Italy
e-mail: johnwillmer.escobar2@unibo.it
Departamento de Ingeniería Civil e Industrial, Pontificia Universidad Javeriana
Calle 18 No. 118-250, 26239 Cali, Colombia
e-mail: jwescobar@javerianacali.edu.co

Rodrigo Linfati
Department of Electronics, Computer Sciences and Systems (DEIS), University of Bologna
Viale Risorgimento 2, 40136 Bologna, Italy
e-mail: rodrigo.linfati@unibo.it

Paolo Toth[1]
Department of Electronics, Computer Sciences and Systems (DEIS), University of Bologna
Viale Risorgimento 2, 40136 Bologna, Italy
e-mail: paolo.toth@unibo.it

ABSTRACT

This paper presents a new metahaheuristic algorithm to solve the capacitated location-routing problem (CLRP). We are given on input a set of identical vehicles, a set of depots with restricted capacities and opening costs, and a set of customers with deterministic demands. The problem consists of determining the depots to be opened, the customers and the vehicles to be assigned to each open depot, and the routes to be performed. The objective is to minimize the sum of the costs of the open depots, of the fixed cost associated with the used vehicles, and of the variable traveling costs related to the performed routes. In the proposed algorithm, a modified granular tabu search with different diversification strategies is applied. Computational experiments on benchmark instances show that the proposed algorithm is able to produce, within short computing time, several solutions obtained by the previously published methods and new best known solutions.

KEYWORDS. Capacitated location-routing problem. Granular tabu search. Metaheuristic algorithms.

MH – Metaheuristic

OC – Combinatorial Optimization

[1] Corresponding author. Fax. +39 051 2093073.

1. Introduction

This paper considers the *Capacitated Location-Routing Problem* (CLRP), i.e. the *Location-Routing Problem* (LRP) with capacity constrains for the depots and the routes. The CLRP can be defined as the following graph theory problem. Let $G = (V, E)$ be a complete undirected graph, in which $V = \{1, \ldots, m + n\}$ is the vertex set and E is the edge set. Vertices $i = 1, \ldots, m$ correspond to the potential depots, each with a capacity W_i and an opening cost O_i. Vertices $j = m + 1, \ldots, m + n$ correspond to the customers, each with a nonnegative demand d_j. A set of homogeneous vehicles, each with capacity Q, is available at each depot. Each vehicle, when is used by a depot to perform a single route, causes a nonnegative fixed cost F. A nonnegative cost c_{ij} is associated with each edge $(i, j) \in E$.

The goal of the CLRP is to determine the depots to be opened, the customers to be assigned to each open depot, and the routes to be performed to fulfill the demand of the customers with the minimum global cost, given by the sum of the costs of the open depots, the costs of the used vehicles, and the costs of the edges traveled by the performed routes. The following constraints are imposed: i) each route must start and finish at the same depot; ii) each customer is visited exactly once by a single route; iii) the sum of the demands of the customers visited by each route must not exceed the vehicle capacity; iv) the sum of the demands of the customers assigned to each depot must not exceed its corresponding capacity.

The CLRP is NP-hard problem since it generalizes three well known NP-hard problems: the *Capacitated Facility Location Problem* (CFLP), the *Capacitated Vehicle Routing Problem* (CVRP), and the *Multi Depot Vehicle Routing Problem* (MDVRP).

A three-index formulation for the CLRP has been introduced by Prins et al. (2007). Two-index formulations have been proposed by Baldacci et al. (2011), Belenguer et al. (2011) and Contardo et al. (2011). These exact approaches have consistently solved to proven optimality instances with less than 100 customers. For this reason, metaheuristic algorithms have been proposed to solve large CLRP instances.

Prins et al. (2007) proposed a two-phase algorithm which exchanges information between phases. In the first phase (location phase), the routes and their customers are aggregated into super customers, and the corresponding capacitated facility location problem is solved by using a Lagrangean relaxation technique. In the second phase (routing phase), a *granular tabu search* (GTS) procedure with one neighborhood was used to solve the resulting multi-depot vehicle routing problem.

A cluster based method for the CLRP was proposed by Barreto et al. (2007). In this work, in the first phase the customer set is split into clusters according to the vehicle capacity. In the second phase, a *Traveling Salesman Problem* (TSP) is solved for each cluster. Finally, in the final phase, the TSP circuits are grouped into super nodes for solving the corresponding capacitated facility location problem.

A metaheuristic for the CLRP has been proposed by Prins et al. (2006a). In this work, a greedy randomized adaptive search procedure (GRASP), with a learning process and a path relinking strategy, has been proposed. A path relinking strategy is then used as post optimization procedure to generate new solutions. The same authors Prins et al. (2006b) proposed a memetic algorithm with population management.

Recently, metaheuristics for solving the CLRP have been developed by Yu et al. (2010), Duhamel et al. (2010) and Hemmelmayr et al. (2011). In the first work, a simulated annealing procedure (SA) based on three random neighborhood structures has been proposed. In the second work, Duhamel et al. (2010) proposed a successful method based on a hybridized GRASP with an evolutionary local search (ELS) procedure. Finally, an adaptive large neighborhood algorithm for the Two- Echelon Vehicle Routing Problem (2E-VRP), which is also able to solve the CLRP, has been introduced by Hemmelmayr et al. (2011).

2. Description of the proposed algorithm

This section presents a new hybrid metaheuristic algorithm (HGTS) developed for solving the CLRP. After the construction of an initial feasible solution by using a *Hybrid procedure*, a *modified Granular Tabu Search* procedure, which considers several diversification steps, is applied to improve the quality of the current solution. Whenever no improvement is obtained within a given number of iterations, the algorithm tries to escape from the current local optimum by applying a randomized *Perturbation procedure*. The proposed approach uses as a general improvement routine a *Procedure VRPH*, based on the library of local search heuristics for the VRP proposed by Groer et al. (2010). The outline of the proposed approach is described in Algorithm 1.

Algorithm 1 Procedure HGTS

```
Input: clrp_instance
Output: clrp_solution

begin
   load  clrp_instance
   clrp_solution = +infinite
   call Hybrid procedure
   clrp_soultion = hybrid_solution
   for i = 0 to Nsplit iterations
       consider the route which contains the longest edge
       split the route
       call LKH heuristic
       call VRPH
       if (split_solution < clrp_solution)
          clrp_solution = split_solution
       endif
    endfor
    call Modified Granular Tabu Search
    update best clrp_solution feasible found so far
    if (clrp_solution = local_minimum)
       call Perturbation
    endif
end
```

The key-point for the success of the proposed algorithm is the correct location of the depots in the *Hybrid procedure.* Since the most critical decisions of the CLRP are those concerning the opening and closing of the depots, a proper location of the depots is able to reduce the search space from a CLRP to a MDVRP. The previously mentioned procedures are described in more detail in the following subsections.

2.1 Procedure VRPH

We developed a procedure, *called VRPH*, which applies three routines proposed by Groer et al. (2010). In particular, the procedure applies routine *vrp_initial* and then, iteratively, routine *vrp_sa* and *vrp_rtr* until no improvement is reached. Basically, *vrp_initial* uses a variant of the Clarke-Wright algorithm to generate initial solutions for the CVRP. The routine *vrp_rtr* is an implementation of the record-to-record travel metaheuristic. Finally, *vrp_sa* is an implementation of a Simulated Annealing (SA) metaheuristic. *Procedure VRPH* is executed in several parts of the proposed approach as a general improvement procedure for a given depot.

2.2 Hybrid procedure

The initial solution S_0 is constructed by using a *Hybrid procedure*, which is able to find good feasible solutions within short computing times. First, a giant TSP tour containing all the customers is constructed by using the well-known Lin-Kernighan heuristic (LKH); for further details see Lin and Kernighan (1973) and Helsgaun et al. (2000). Then, the giant TSP tour is split into several clusters so as to satisfy the route capacity constraints.

For each depot i and each cluster g a TSP tour is obtained by using procedure LKH to evaluate the traveling cost between i and g. Then, we assign the depots to the clusters by solving an ILP model corresponding to the formulation of the well-known *Single Source Capacitated Location Problem;* for further details see e.g. Barcelo & Casanovas (1984), and Klincewicz & Luss (1986).

Note that the initial solution procedure is repeated n times by selecting each customer as initial vertex to split the giant tour. Finally, a splitting procedure is applied to reduce the traveling cost by adding new routes, and by assigning them to different depots; for further details see Escobar et al. (2011). We repeat the Splitting procedure *Nsplit* times (where *Nsplit* is a given parameter), by considering at each iteration a different route.

2.3 Modified Granular Tabu Search

In this stage, the algorithm tries to improve the initial solution S_0 obtained by the *Hybrid procedure* applying a *modified Granular Tabu Search* (GTS) procedure. The goal of the modified GTS is to improve the routes without considering moves between close and open depots; hence the search space is related to a MDVRP.

The *granular tabu search* (GTS) approach has been proposed by Toth and Vigo (2003). The method is based on the use of a sparse graph, which drastically reduces the time required by a tabu search algorithm. In particular, the original complete graph G is replaced by a sparse graph which includes all the edges whose cost is smaller than the *granularity threshold* ϑ, the edges incident to the depot, and those belonging to the best solution found so far. The value of ϑ is defined by means of an increasing function of the *sparsification factor* β: $\vartheta = \beta z^*$, where z^* is the average cost of the edges in the current best solution found so far.

The main objective of the GTS approach is to have good solutions within short computing times. Three main differences with respect to the idea of "granularity" introduced by Toth and Vigo (2003) for the CVRP are considered here. Basically, the proposed algorithm considers five *neighborhoods (*Insertion, Swap, Two Opt, Exchange, and Inter-tour exchange), three different *diversification strategies*, and a *random perturbation procedure* to avoid that the algorithm remains in a local optimum for a given number of iterations. The proposed *diversification strategies* and *perturbation procedure* are described in the following subsections.

2.4 Diversification strategies

Three diversification strategies have been considered. The first strategy is based on the granularity diversification proposed in Toth and Vigo (2003). Initially, the sparsification factor β is set to an initial value β_0. If no improvement of the best feasible solution found so far is reached after a given number of iterations, the sparsification factor β is increased to a β_d value. A new sparse graph is then calculated, and $N_{moviter}$ iterations are executed starting from the best solution found so far. Finally, the sparsification factor β is reset to its initial value β_0 and the search continues.

The second strategy is based on a penalty approach. In particular, the proposed approach allows infeasible solutions with respect to the depot and the vehicle capacities. Given a feasible solution S, we assign to its objective function $F_1(S)$ a value equal to the sum of the opening costs of the open depots, of the traveling costs of the edges belonging to the routes traversed by S, and of the fixed costs of the vehicles used in S. In addition, for any solution S infeasible with respect to the depot capacity, we add to $F_1(S)$ a penalty term obtained by multiplying the over depot capacity by a dynamically changing penalty factor P_d. Consequently, the objective function $F_2(S)$ is obtained. A similar approach is used to calculate the objective function value of any solution S infeasible with respect to the route capacity by using a penalty factor P_r. Note that if the solution S is feasible $F_2(S) = F_1(S)$.

In the selection of the best move to be performed we introduce an extra penalty by adding to $F_2(S)$ a constant term equal to the product of the absolute difference value Δ_{max} between two successive values of the objective function, the square root of the number of routes k, and a scaling factor g; for further details see Taillard (1993).

Finally, the third diversification strategy determines every Ng iterations a random feasible solution for each open depot by using *procedure VRPH*.

2.5 Perturbation procedure

Since the modified GTS procedure can fail in finding a move improving the current solution, the algorithm tries to escape from a local optimum by perturbing the current solution. In particular, if no improving move has been performed after N_{pert} iterations, the algorithm applies a perturbation approach similar to the "3-route procedure" proposed by Renaud et al. (1996); differently from what is proposed by this work, we consider a randomized procedure for selecting the routes to be perturbed.

3. Computational results

The overall algorithm (HGTS) has been implemented in C++, and the computational experiments have been performed on an Intel Core Duo CPU (2.00 GHz) under Linux Ubuntu 11.04 with 2 GB of memory. The performance of the proposed algorithm has been evaluated by considering 79 benchmark instances taken from the literature. The complete set of instances considers three data subsets. The first data subset (DS1) was proposed by Tuzun and Burke (1999) and considers 36 instances with capacity constraints only on the routes. It considers instances with $n =$ 100, 150 and 200 customers. The number m of potential depots is either 10 or 20. The vehicle capacity Q is set to 150, and the demands of the customers are uniformly random distributed in the interval [1, 20].

The second data subset (DS2) was proposed by Prins et al. (2004), and contains 30 instances with capacity constraints on both the routes and the depots. The number m of potential depots is either 5 or 10, and the number of customers is $n =$ 20, 50, 100 and 200. The vehicle capacity Q is either 70 or 150, and the demands of the customers are uniformly random distributed in the interval [11, 20].

The instances of the third data subset (DS3), introduced by Barreto (2004), were obtained from some classical CVRP instances by adding new depots with the corresponding capacities and fixed costs. This data subset considers 19 instances, but generally only 13 instances have been tested by the previous heuristics algorithms. The number of customers ranges from 21 to 150, and the number of potential depots from 5 to 10.

For each instance, only one run of the proposed algorithm is executed. As for other heuristics, extensive computational tests have been made to find a suitable set of parameters. The proposed algorithm has been compared (see Tables 1-5) with five effective published metaheuristics proposed for the CLRP: GRASP of Prins et al. (2006a), the memetic algorithm with population management (MA|PM) of Prins et al. (2006b), the Langrangean relaxation and granular tabu search method (LRGTS) of Prins et al. (2007), GRASP+ELS of Duhamel et al. (2010), the simulated annealing algorithm (SALRP) of Yu et al. (2010), the Adaptive Large Neighborhood Search (ALNS) of Hemmelmayr et al. (2011). The results reported for GRASP, MA|PM, LGRTS and SALRP correspond to a single run of the associated algorithm. GRASP+ELS and ALNS have been executed five times. For GRASP+ELS has been considered five different random generator seeds, and the reported cost is the best found over the five runs. In addition, the reported computing time is the time required to reach the best solution within the corresponding run. The reported results for ALNS correspond to the average solutions over the runs and the complete running time. In the paper by Yu et al. (2010), the authors report also the cost of the best solution found by SALRP during the parameter analysis phase. In Tables 1 to 5, the following notation is used:

Instance	name of instance;
n	number of customers;
m	number of potential depots;
BKC	cost of the best result among the algorithms;
BKS	cost of the best-known result obtained either by the considered algorithms (BKC) or during the parameter analysis phase of SALRP;
CPU	CPU used by each method;
CPU index	Passmark performance test for each CPU;
CPU time	running time in seconds on the CPU used by each algorithm;
Gap BKC	percentage gap of the solution cost found by each algorithm with respect to BKC;
Gap BKS	percentage gap of the solution cost found by each algorithm with respect to BKS.

The CPU index is given by the Passmark performance test. This is a well-known benchmark test focused on CPU and memory performance. Higher values of the Passmark test indicate that the corresponding CPU is faster.

A summary about the results obtained by the considered six algorithms for the complete instance dataset is given in Tables 1and 2. Table 1 provides the average values of Gap BKS, Gap BKC and CPU time, and the CPU index of the corresponding CPU. Table 2 reports the number of BKC solutions obtained by each algorithm. Table 1 shows that the proposed algorithm reports better global average results respect to Gap BKS and Gap BKC than those obtained by GRASP, MA|PM, LRGTS, GRASP+ELS and SALRP. Only ALNS is able to obtain a slightly better average values (global average of Gap BKS and Gap BKC), although with large CPU times. As for the global CPU time, the proposed algorithm is faster than GRASP+ELS, ALNS and SALRP, which were able to find the previous best results in terms of average gaps and number of best solutions. It is to note that the CPU time reported for algorithm GRASP+ELS does not represent the global time required to find the best solution (obtained by executing five runs), since it corresponds to the CPU time spent, for each instance, in a single run. On the other hand, the CPU time of HGTS is larger than that of GRASP, MA|PM and LGRTS. This can be explained by the fact that we use several improvement procedures in the second phase. Although the CPU time of the proposed algorithm is larger than that of these approaches, it remains within an acceptable range for a strategic problem like CLRP. In addition, algorithm HGTS is able to find the largest number of BKC. The detailed results for the first, the second and the third data subset are shown in Tables 4, 5 and 6 respectively.

4. Concluding remarks

We propose an effective hybrid metaheuristic algorithm for the capacitated location routing problem (CLRP). In the proposed heuristic, after the construction of an initial feasible solution, we apply a modified Granular Tabu Search which considers five granular neighborhoods, three different diversification strategies and a perturbation procedure. The perturbation procedure is applied whenever the algorithm remains in a local optimum for a given number of iterations. We compared the proposed algorithm with five most effective published metaheuristics for the CLRP on a set of benchmark instances from the literature. The results show the effectiveness of the proposed algorithm, and several best known solutions are improved within reasonable computing times. The results obtained suggest that the proposed framework could be applied to other problems as the periodic location-routing problem (PLRP), the multi depot vehicle routing problem (MDVRP) and several extensions of the CLRP obtained by adding constraints as time windows, heterogeneous fleet, etc.

Acknowledgments

The work of the first author has been partially supported by MIUR (Ministero Istruzione, Università e Ricerca), Italy and by Pontificia Universidad Javeriana, Cali, Colombia. This support is gratefully acknowledged. The authors are extremely grateful to anonymous referees for the very helpful comments.

References

Baldacci R., Mingozzi A., and Wolfler-Calvo R. (2011), An Exact Method for the Capacitated Location-Routing Problem, *Operations Research*, 59 (5), 1284-1296**.**

Barcelo J. and Casanovas J. (1984), A heuristic Lagrangean algorithm for the capacitated plant location problem, *European Journal of Operational Research*, 15 (2), 212-226.

Barreto S. (2004), Análise e Modelização de Problemas de localização-distribuição (Analysis and modelling of location-routing problems), *PhD thesis, University of Aveiro*, 3810-4193.

Barreto S., Ferreira C., Paixao J. and Santos B.S. (2007), Using clustering analysis in a capacitated location-routing problem, *European Journal of Operational Research*, 179 (1), 968-977.

Belenguer J.M., Benavent E., Prins C., Prodhon C. and Wolfler-Calvo R. (2011), A Branch-and-Cut method for the Capacitated Location-Routing Problem, *Computers and Operations Research*, 38 (6), 931-941.

Contardo C., Cordeau J.F. and Gendron B. (2011), A branch-and-cut-and-price algorithm for the capacitated location-routing problem. *Technical Report*, CIRRELT, Canada.

Duhamel C., Lacomme P., Prins C. and Prodhon C. (2010), A GRASP x ELS approach for the capacitated location-routing problem, *Computers and Operations Research*, (37) 11, 1912-1923.

Groer, C., Golden B. and Wasil E. (2010), A library of local search heuristics for the vehicle routing problem, *Mathematical Programming Computation*, 2 (2), 79-101.

Escobar J.W, Linfati R and Toth P. (2011). A two-phase hybrid heuristic algorithm for the capacitated location-routing problem. *Technical Report*, DEIS, University of Bologna, Italy.

Helsgaun K. (2000), An effective implementation of the Lin-Kernighan traveling salesman heuristic, *European Journal of Operational Research*, 126 (1), 106-130.

Hemmelmayr V., Cordeau J., Crainic T. (2011), An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics, *Technical Report*, Université de Montréal, 2011.

Klincewicz J.G. and Luss H. (1986), A Lagrangian relaxation heuristic for capacitated facility location with single-source constraints, *Journal of the Operational Research Society*, 37 (5), 495-500.
Lin S. and Kernighan B.W. (1973), An effective heuristic algorithm for the traveling-salesman problem, *Operations Research*, 21 (2), 498-516.

Prins C., Prodhon C. and Wolfler-Calvo R. (2004), Nouveaux algorithmes pour le problème de localisation et routage sous contraintes de capacité, MOSIM, 4, 1115-1122.

Prins C., Prodhon C. and Wolfler-Calvo R. (2006a), Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking, *4OR: A Quarterly Journal of Operations Research*, 4 (3), 221-238.

Prins C., Prodhon C. and Wolfler-Calvo R. (2006b), A memetic algorithm with population management (MA| PM) for the capacitated location-routing problem", *Lecture Notes in Computer Science*, 3906, 183-194.

Prins C., Prodhon C., Ruiz A., Soriano P. and Wolfler-Calvo R. (2007), Solving the capacitated location-routing problem by a cooperative Lagrangean relaxation-granular tabu search heuristic, *Transportation Science*, 41 (4), 470-483.

Taillard É. (1993), Parallel iterative search methods for vehicle routing problems, *Networks*, 23 (8), 661-673.

Toth P. and Vigo D. (2003), The granular tabu search and its application to the vehicle-routing problem, *INFORMS Journal on Computing*, 15 (4), 333-346.

Tuzun D. and Burke L.I. (1999), A two-phase tabu search approach to the location routing problem, *European Journal of Operational Research*, 116 (1), 87-99.

Yu V.F., Lin S.W., Lee W. and Ting C.J. (2010), A simulated annealing heuristic for the capacitated location routing problem, *Computers and Industrial Engineering*, 58 (2), 288-299.

Table 1. Summarized results on GAP BKS, GAP BKC and CPU time for the complete data set

	Size	GRASP [2006]			MA|PM [2006]			LRGTS [2007]			GRASP + ELS [2010]			SALRP [2010]			ALNS [2011]			HGTS [2011]		
		Gap BKS	Gap BKC	CPU time			CPU time		Gap BKC	CPU time	Gap BKS			Gap BKS	Gap BKC	CPU time	Gap BKS	Gap BKC	CPU time	Gap BKS		CPU time
DS1	36	3.07	2.93	163	1.44	1.31	207	1.42	1.29	22	0.87	0.74	607	1.07	0.94	826	**0.47**	**0.34**	830	0.72	0.59	392
DS2	30	3.57	3.45	97	1.35	1.23	96	0.71	0.59	18	1.04	0.92	258	**0.38**	**0.27**	422	0.65	0.53	451	0.49	0.38	176
DS3	13	1.63	1.58	20	2.06	2.01	36	1.66	1.61	18	**0.08**	**0.03**	188	0.29	0.25	161	0.25	0.20	177	0.78	0.74	105
		3.02	2.91	114	1.51	1.39	137	1.19	1.08	20	0.81	0.69	405	0.68	0.57	564	**0.50**	**0.39**	579	0.65	0.54	263
						Intel Pentium 4 (2.40 Ghz)			Intel Pentium 4 (2.40 Ghz)			Intel Core2 Quad (2.83 Ghz)						AMD Opt. 275 (2.20 Ghz)			Intel Core2 Duo (2.00 Ghz)	
CPU index			314			314			314			4373			4046			1234			1398	

Table 2. Summarized results on the number of BKC for the complete data set

	GRASP [2006]	MA|PM [2006]	LRGTS [2007]	GRASP+ELS [2010]	SALRP [2010]	ALNS [2011]	HGTS [2011]
DS1 (36 Instances)							
Total BKC	0	1	0	12	7	8	**16**
DS2 (30 Instances)							
Total BKC	4	11	6	13	**14**	8	**14**
DS2 (13 Instances)							
Total BKC	4	5	2	**11**	**11**	9	8
BKC overall	8	17	8	36	32	25	**38**

Table 3. Detailed results for the first data subset DS1 (Tuzun-Burke Instances)

Instance	n	m	BKS	BKC	GRASP [2006]				MA|PM [2006]				LRGTS [2007]				GRASP + ELS [2010]				SALRP [2010]				ALNS [2011]				HGTS [2011]			
					Cost	Gap BKS	Gap BKC	CPU time	Cost	Gap BKS	Gap BKC	CPU time	Cost	Gap BKS	Gap BKC	CPU time	Cost	Gap BKS	Gap BKC	CPU time	Cost	Gap BKS	Gap BKC	CPU time	Cost	Gap BKS	Gap BKC	CPU time	Cost	Gap BKS	Gap BKC	CPU time
111112	100	10	1467.68	1473.36	1525.25	3.92	3.52	33	1493.92	1.79	1.40	33	1490.82	1.58	1.19	3	**1473.36**	0.39	0.00	233	1477.24	0.65	0.26	369	1475.67	0.54	0.16	275	1479.21	0.79	0.40	152
111122	100	20	1449.20	1449.20	1526.90	5.36	5.36	41	1471.36	1.53	1.53	36	1471.76	1.56	1.56	8	**1449.20**	0.00	0.00	9	1470.96	1.50	1.50	274	1464.72	1.07	1.07	321	1486.27	2.56	2.56	239
111212	100	10	1394.80	1396.59	1423.54	2.06	1.93	28	1418.83	1.72	1.59	36	1412.04	1.24	1.11	4	**1396.59**	0.13	0.00	112	1408.65	0.99	0.86	231	1400.49	0.41	0.28	244	1407.26	0.89	0.76	120
111222	100	20	1432.29	1432.29	1482.29	3.49	3.49	36	1492.46	4.20	4.20	36	1443.06	0.75	0.75	8	**1432.29**	0.00	0.00	114	**1432.29**	0.00	0.00	420	1441.21	0.62	0.62	376	1474.01	2.91	2.91	146
112112	100	10	1167.16	1167.16	1200.24	2.83	2.83	28	1173.22	0.52	0.52	33	1187.63	1.75	1.75	8	**1167.16**	0.00	0.00	27	1177.14	0.86	0.86	348	1173.04	0.50	0.50	489	**1167.16**	0.00	0.00	232
112122	100	20	1102.24	1102.24	1123.64	1.94	1.94	34	1115.37	1.19	1.19	43	1115.95	1.24	1.24	8	**1102.24**	0.00	0.00	259	1110.36	0.74	0.74	342	1102.34	0.01	0.01	373	**1102.24**	0.00	0.00	224
112212	100	10	791.66	791.66	814.00	2.82	2.82	23	793.97	0.29	0.29	38	813.28	2.73	2.73	5	792.03	0.05	0.05	5	**791.66**	0.00	0.00	360	791.83	0.02	0.02	739	**791.66**	0.00	0.00	201
112222	100	20	728.30	728.30	747.84	2.68	2.68	38	730.51	0.30	0.30	49	742.96	2.01	2.01	6	**728.30**	0.00	0.00	48	731.95	0.50	0.50	418	728.32	0.00	0.00	384	**728.30**	0.00	0.00	254
113112	100	10	1238.49	1238.49	1273.10	2.79	2.79	23	1262.32	1.92	1.92	38	1267.93	2.38	2.38	4	1240.39	0.15	0.15	55	**1238.49**	0.00	0.00	300	1240.31	0.15	0.15	357	**1238.49**	0.00	0.00	160
113122	100	20	1245.31	1246.00	1272.94	2.22	2.16	36	1251.32	0.48	0.43	48	1256.12	0.87	0.81	6	**1246.00**	0.06	0.00	233	1247.28	0.16	0.10	428	1248.17	0.23	0.17	445	1251.22	0.47	0.42	237
113212	100	10	902.26	902.26	912.19	1.10	1.10	20	903.82	0.17	0.17	35	913.06	1.20	1.20	4	902.30	0.00	0.00	249	**902.26**	0.00	0.00	291	902.27	0.00	0.00	321	**902.26**	0.00	0.00	135
113222	100	20	1018.29	1018.29	1022.51	0.41	0.41	38	1022.93	0.46	0.46	63	1025.51	0.71	0.71	5	**1018.29**	0.00	0.00	196	1024.02	0.56	0.56	316	1018.56	0.03	0.03	386	**1018.29**	0.00	0.00	157
Avg.						**2.64**	**2.59**	**32**		**1.22**	**1.17**	**41**		**1.50**	**1.45**	**6**		**0.06**	**0.02**	**128**		**0.50**	**0.45**	**341**		**0.30**	**0.25**	**393**		**0.64**	**0.59**	**188**
131112	150	10	1922.59	1939.52	2006.70	4.37	3.46	113	1959.39	1.91	1.02	129	1946.01	1.22	0.33	13	1944.57	1.14	0.26	518	1953.85	1.63	0.74	743	**1939.52**	0.88	0.00	504	1961.75	2.04	1.15	485
131122	150	20	1833.95	1856.51	1888.90	3.00	1.74	161	1881.67	2.60	1.36	144	1875.79	2.28	1.04	19	1864.24	1.65	0.42	705	1899.05	3.55	2.29	835	1857.29	1.27	0.04	635	**1856.51**	1.23	0.00	298
131212	150	10	1978.27	1984.25	2033.93	2.81	2.50	100	**1984.25**	0.30	0.00	111	2010.53	1.63	1.32	11	1992.41	0.71	0.41	727	2057.53	4.01	3.69	456	2009.44	1.58	1.27	664	2012.69	1.74	1.43	406
131222	150	20	1801.39	1801.39	1856.07	3.04	3.04	133	1855.25	2.99	2.99	144	1819.89	1.03	1.03	16	1835.25	1.88	1.88	415	**1801.39**	0.00	0.00	833	1838.51	2.06	2.06	485	1803.01	0.09	0.09	302
132112	150	10	1445.25	1445.25	1508.33	4.36	4.36	118	1448.27	0.21	0.21	168	1448.65	0.24	0.24	23	1453.78	0.59	0.59	103	1453.30	0.56	0.56	750	1449.15	0.27	0.27	1049	**1445.25**	0.00	0.00	449
132122	150	20	1441.98	1444.17	1456.82	1.03	0.88	166	1459.83	1.24	1.08	155	1492.86	3.53	3.37	28	**1444.17**	0.15	0.00	662	1455.50	0.94	0.78	828	1446.91	0.34	0.19	805	1452.07	0.70	0.55	493
132212	150	10	1204.42	1204.42	1240.40	2.99	2.99	134	1207.41	0.25	0.25	201	1211.07	0.55	0.55	19	1219.86	1.28	1.28	459	1206.24	0.15	0.15	752	1205.83	0.12	0.12	2197	**1204.42**	0.00	0.00	270
132222	150	20	930.99	931.49	940.80	1.05	1.00	143	934.79	0.41	0.35	196	936.93	0.64	0.58	14	945.81	1.59	1.54	224	934.62	0.39	0.34	842	933.14	0.23	0.18	982	**931.49**	0.05	0.00	335
133112	150	10	1700.39	1700.39	1736.90	2.15	2.15	93	1720.30	1.17	1.17	144	1729.31	1.70	1.70	18	1712.11	0.69	0.69	271	1720.81	1.20	1.20	742	**1700.39**	0.00	0.00	1046	1705.36	0.29	0.29	444
133122	150	20	1400.01	1402.94	1425.74	1.84	1.63	128	1429.34	2.09	1.88	156	1424.59	1.76	1.54	19	**1402.94**	0.21	0.00	524	1415.85	1.13	0.92	833	1403.50	0.25	0.04	925	1416.74	1.19	0.98	342
133212	150	10	1199.27	1199.27	1223.70	2.04	2.04	89	1203.44	0.35	0.35	154	1216.32	1.42	1.42	15	1214.82	1.30	1.30	251	1216.84	1.47	1.47	756	**1199.27**	0.00	0.00	1375	1234.83	2.97	2.97	526
133222	150	20	1152.18	1154.36	1231.33	6.87	6.67	135	1158.54	0.55	0.36	223	1162.16	0.87	0.68	14	1155.96	0.33	0.14	375	1159.12	0.60	0.41	837	**1154.36**	0.19	0.00	911	1156.05	0.34	0.15	380
Avg.						**2.96**	**2.70**	**126**		**1.17**	**0.92**	**160**		**1.40**	**1.15**	**17**		**0.96**	**0.71**	**436**		**1.30**	**1.05**	**767**		**0.60**	**0.35**	**965**		**0.89**	**0.63**	**394**
121112	200	10	2265.59	2265.59	2384.01	5.23	5.23	385	2293.99	1.25	1.25	523	2296.52	1.37	1.37	41	2295.90	1.34	1.34	655	2324.10	2.58	2.58	1328	2278.27	0.56	0.56	944	**2265.59**	0.00	0.00	522
121122	200	20	2166.43	2166.43	2288.09	5.62	5.62	410	2277.39	5.12	5.12	458	2207.50	1.90	1.90	40	2203.57	1.71	1.71	432	2258.16	4.23	4.23	1455	2192.61	1.21	1.21	847	**2166.43**	0.00	0.00	603
121212	200	10	2245.33	2246.39	2273.19	1.24	1.19	311	2274.57	1.30	1.25	378	2260.87	0.69	0.64	33	**2246.39**	0.05	0.00	1566	2260.30	0.67	0.62	1319	2247.75	0.11	0.06	907	2249.40	0.18	0.13	527
121222	200	20	2237.81	2237.81	2345.10	4.79	4.79	419	2376.25	6.19	6.19	436	2259.52	0.97	0.97	40	2265.53	1.24	1.24	2192	2326.53	3.96	3.96	1428	2263.20	1.13	1.13	860	**2237.81**	0.00	0.00	558
122112	200	10	2089.77	2093.78	2137.08	2.26	2.07	338	2106.26	0.79	0.60	351	2120.76	1.48	1.29	48	2106.47	0.80	0.61	1521	2112.65	1.09	0.90	1320	**2093.78**	0.19	0.00	1606	2121.93	1.54	1.34	522
122122	200	20	1719.96	1722.99	1807.29	5.08	4.89	370	1771.53	3.00	2.82	378	1737.81	1.04	0.86	59	1779.05	3.44	3.25	618	**1722.99**	0.18	0.00	1400	1732.00	0.70	0.52	941	1749.10	1.69	1.52	691
122212	200	10	1462.15	1462.15	1496.75	2.37	2.37	243	1467.54	0.37	0.37	323	1488.55	1.81	1.81	38	1474.25	0.83	0.83	514	1469.10	0.48	0.48	1299	**1462.15**	0.00	0.00	1861	1473.27	0.76	0.76	724
122222	200	20	1082.59	1082.59	1095.92	1.23	1.23	309	1088.00	0.50	0.50	505	1090.59	0.74	0.74	39	1085.69	0.29	0.29	1243	1088.64	0.56	0.56	1429	1086.08	0.32	0.32	812	**1082.59**	0.00	0.00	616
123112	200	10	1970.44	1971.01	2044.66	3.77	3.74	283	1973.28	0.14	0.12	413	1984.06	0.69	0.66	43	2004.33	1.72	1.69	1451	1994.16	1.20	1.17	1318	**1971.01**	0.03	0.00	968	1984.77	0.73	0.70	542
123122	200	20	1918.93	1932.05	2090.95	8.96	8.22	399	1979.05	3.13	2.43	406	1986.49	3.52	2.82	53	1964.40	2.37	1.67	1273	**1932.05**	0.68	0.00	1412	1952.31	1.74	1.05	740	1958.98	2.09	1.39	617
123212	200	10	1764.16	1764.16	1788.70	1.39	1.39	199	1782.23	1.02	1.02	353	1786.79	1.28	1.28	34	1778.80	0.83	0.83	1398	1779.10	0.85	0.85	1314	**1764.16**	0.00	0.00	2055	1778.41	0.81	0.81	697
123222	200	20	1390.87	1390.87	1408.63	1.28	1.28	296	1396.24	0.39	0.39	530	1401.16	0.74	0.74	43	1453.82	4.53	4.53	2202	1396.42	0.40	0.40	1427	1395.38	0.32	0.32	1038	**1390.87**	0.00	0.00	518
Avg.						**3.60**	**3.50**	**330**		**1.93**	**1.84**	**421**		**1.35**	**1.26**	**43**		**1.59**	**1.50**	**1255**		**1.41**	**1.31**	**1371**		**0.53**	**0.43**	**1132**		**0.65**	**0.55**	**595**
Global Avg.						**3.07**	**2.93**	**163**		**1.44**	**1.31**	**207**		**1.42**	**1.29**	**22**		**0.87**	**0.74**	**607**		**1.07**	**0.94**	**826**		**0.47**	**0.34**	**830**		**0.72**	**0.59**	**392**

Table 4. Detailed results for the second data subset DS2 (Prodhon Instances)

Instance	n	m	BKS	BKC	GRASP [2006]				MA|PM [2006]				LRGTS [2007]				GRASP + ELS [2010]				SALRP [2010]				ALNS [2011]				HGTS [2011]			
					Cost	Gap BKS	Gap BKC	CPU time	Cost	Gap BKS	Gap BKC	CPU time	Cost	Gap BKS	Gap BKC	CPU time	Cost	Gap BKS	Gap BKC	CPU time	Cost	Gap BKS	Gap BKC	CPU time	Cost	Gap BKS	Gap BKC	CPU time	Cost	Gap BKS	Gap BKC	CPU time
20-5-1a	20	5	54793	54793	55021	0.42	0.42	0	**54793**	0.00	0.00	0	55131	0.62	0.62	1	**54793**	0.00	0.00	0	**54793**	0.00	0.00	20	**54793**	0.00	0.00	39	**54793**	0.00	0.00	3
20-5-1b	20	5	39104	39104	**39104**	0.00	0.00	0	**39104**	0.00	0.00	0	**39104**	0.00	0.00	0	**39104**	0.00	0.00	0	**39104**	0.00	0.00	15	**39104**	0.00	0.00	54	**39104**	0.00	0.00	4
20-5-2a	20	5	48908	48908	**48908**	0.00	0.00	0	**48908**	0.00	0.00	1	**48908**	0.00	0.00	1	**48908**	0.00	0.00	0	**48908**	0.00	0.00	19	**48908**	0.00	0.00	38	48945	0.08	0.08	3
20-5-2b	20	5	37542	37542	**37542**	0.00	0.00	0	**37542**	0.00	0.00	0	**37542**	0.00	0.00	0	**37542**	0.00	0.00	0	**37542**	0.00	0.00	15	**37542**	0.00	0.00	67	**37542**	0.00	0.00	4
Avg.						**0.10**	**0.10**	**0**		**0.00**	**0.00**	**0**		**0.15**	**0.15**	**1**		**0.00**	**0.00**	**0**		**0.00**	**0.00**	**17**		**0.00**	**0.00**	**50**		**0.02**	**0.02**	**4**
50-5-1a	50	5	90111	90111	90632	0.58	0.58	2	90160	0.05	0.05	4	90160	0.05	0.05	0	**90111**	0.00	0.00	3	**90111**	0.00	0.00	75	**90111**	0.00	0.00	101	90402	0.32	0.32	27
50-5-1b	50	5	63242	63242	64761	2.40	2.40	2	**63242**	0.00	0.00	5	63256	0.02	0.02	1	**63242**	0.00	0.00	0	**63242**	0.00	0.00	58	**63242**	0.00	0.00	65	64073	1.31	1.31	27
50-5-2a	50	5	88298	88298	88786	0.55	0.55	3	**88298**	0.00	0.00	5	88715	0.47	0.47	2	88643	0.39	0.39	11	**88298**	0.00	0.00	95	88576	0.31	0.31	99	89342	1.18	1.18	23
50-5-2b	50	5	67308	67308	68042	1.09	1.09	3	67893	0.87	0.87	5	67698	0.58	0.58	2	**67308**	0.00	0.00	16	67340	0.05	0.05	59	67448	0.21	0.21	200	68479	1.74	1.74	21
50-5-2bis	50	5	84055	84055	**84055**	0.00	0.00	2	**84055**	0.00	0.00	5	84181	0.15	0.15	3	**84055**	0.00	0.00	0	**84055**	0.00	0.00	75	84119	0.08	0.08	107	**84055**	0.00	0.00	23
50-5-2bbis	50	5	51822	51822	52059	0.46	0.46	3	**51822**	0.00	0.00	6	51992	0.33	0.33	1	**51822**	0.00	0.00	11	**51822**	0.00	0.00	66	51840	0.03	0.03	98	52087	0.51	0.51	29
50-5-3a	50	5	86203	86203	87380	1.37	1.37	3	**86203**	0.00	0.00	5	**86203**	0.00	0.00	0	**86203**	0.00	0.00	0	86456	0.29	0.29	74	86262	0.07	0.07	101	**86203**	0.00	0.00	66
50-5-3b	50	5	61830	61830	61890	0.10	0.10	3	**61830**	0.00	0.00	8	**61830**	0.00	0.00	1	**61830**	0.00	0.00	0	62700	1.41	1.41	58	**61830**	0.00	0.00	137	**61830**	0.00	0.00	38
Avg.						**0.82**	**0.82**	**3**		**0.12**	**0.12**	**5**		**0.20**	**0.20**	**1**		**0.05**	**0.05**	**5**		**0.22**	**0.22**	**70**		**0.09**	**0.09**	**114**		**0.63**	**0.63**	**32**
100-5-1a	100	5	275419	276186	279437	1.46	1.18	28	281944	2.37	2.08	33	277935	0.91	0.63	9	276960	0.56	0.28	148	277035	0.59	0.31	349	276364	0.34	0.06	520	**276186**	0.28	0.00	157
100-5-1b	100	5	213615	214885	216159	1.19	0.59	24	216656	1.42	0.82	44	**214885**	0.59	0.00	9	215854	1.05	0.45	68	216002	1.12	0.52	269	215059	0.68	0.08	1190	214892	0.60	0.00	136
100-5-2a	100	5	193671	193903	199520	3.02	2.90	18	195568	0.98	0.86	45	196545	1.48	1.36	3	194267	0.31	0.19	212	194124	0.23	0.11	349	**193903**	0.12	0.00	463	194625	0.49	0.37	145
100-5-2b	100	5	157150	157150	159550	1.53	1.53	23	157325	0.11	0.11	45	157792	0.41	0.41	4	157375	0.14	0.14	125	**157150**	0.00	0.00	212	157157	0.00	0.00	859	157319	0.11	0.11	193
100-5-3a	100	5	200079	200242	203999	1.96	1.88	21	201749	0.83	0.75	36	201952	0.94	0.85	3	200345	0.13	0.05	141	**200242**	0.08	0.00	250	200496	0.21	0.13	454	201086	0.50	0.42	163
100-5-3b	100	5	152441	152467	154596	1.41	1.40	20	153322	0.58	0.56	43	154709	1.49	1.47	3	152528	0.06	0.04	221	**152467**	0.02	0.00	197	152900	0.30	0.28	684	153663	0.80	0.78	168
Avg.						**1.76**	**1.58**	**22**		**1.05**	**0.87**	**41**		**0.97**	**0.79**	**5**		**0.37**	**0.19**	**153**		**0.34**	**0.16**	**271**		**0.28**	**0.09**	**695**		**0.46**	**0.28**	**160**
100-10-1a	100	10	287983	289755	323171	12.22	11.53	38	316575	9.93	9.26	31	291887	1.36	0.74	14	301418	4.67	4.03	48	291043	1.06	0.44	270	299982	4.17	3.53	210	**289755**	0.62	0.00	277
100-10-1b	100	10	231763	234210	271477	17.14	15.91	30	270251	16.61	15.39	45	235532	1.63	0.56	14	269594	16.32	15.11	186	**234210**	1.06	0.00	203	240829	3.91	2.83	188	238002	2.69	1.62	152
100-10-2a	100	10	243590	243778	254087	4.31	4.23	39	245123	0.63	0.55	31	246708	1.28	1.20	15	**243778**	0.08	0.00	260	245813	0.91	0.83	261	245548	0.80	0.73	136	245768	0.89	0.82	92
100-10-2b	100	10	203988	203988	206555	1.26	1.26	30	205052	0.52	0.52	39	204435	0.22	0.22	10	**203988**	0.00	0.00	139	205312	0.65	0.65	199	204494	0.25	0.25	261	204252	0.13	0.13	99
100-10-3a	100	10	250882	250882	270826	7.95	7.95	35	253669	1.11	1.11	36	258656	3.10	3.10	14	253511	1.05	1.05	164	**250882**	0.00	0.00	338	254882	1.59	1.59	202	254716	1.53	1.53	125
100-10-3b	100	10	204317	204815	216173	5.80	5.55	40	**204815**	0.24	0.00	45	205883	0.77	0.52	11	205087	0.38	0.13	203	205009	0.34	0.09	240	206175	0.91	0.66	224	205837	0.74	0.50	144
Avg.						**8.11**	**7.74**	**35**		**4.84**	**4.47**	**38**		**1.39**	**1.06**	**13**		**3.75**	**3.39**	**167**		**0.67**	**0.34**	**252**		**1.94**	**1.60**	**204**		**1.10**	**0.77**	**148**
200-10-1a	200	10	476778	476778	490820	2.95	2.95	518	483497	1.41	1.41	431	481676	1.03	1.03	63	486467	2.03	2.03	1521	481002	0.89	0.89	1428	483205	1.35	1.35	752	**476778**	0.00	0.00	671
200-10-1b	200	10	378289	378289	416753	10.17	10.17	379	380044	0.46	0.46	579	380613	0.61	0.61	60	382329	1.07	1.07	359	383586	1.40	1.40	1336	380538	0.59	0.59	1346	**378289**	0.00	0.00	476
200-10-2a	200	10	449849	449951	512679	13.97	13.94	554	451840	0.44	0.42	351	453353	0.78	0.76	60	452276	0.54	0.52	112	450848	0.22	0.20	1796	451750	0.42	0.40	1201	**449951**	0.02	0.00	483
200-10-2b	200	10	374330	374961	379980	1.51	1.34	368	375019	0.18	0.02	401	377351	0.81	0.64	78	376027	0.45	0.28	1610	376674	0.63	0.46	1245	376112	0.48	0.31	1349	**374961**	0.17	0.00	530
200-10-3a	200	10	472321	472321	496694	5.16	5.16	425	478132	1.23	1.23	266	476684	0.92	0.92	78	478380	1.28	1.28	1596	473875	0.33	0.33	1776	479366	1.49	1.49	1251	**472321**	0.00	0.00	624
200-10-3b	200	10	362817	363252	389016	7.22	7.09	290	364834	0.56	0.44	341	365250	0.67	0.55	74	365166	0.65	0.53	591	363701	0.24	0.12	1326	366902	1.13	1.00	1137	**363252**	0.12	0.00	389
Avg.						**6.83**	**6.77**	**422**		**0.71**	**0.66**	**395**		**0.80**	**0.75**	**69**		**1.00**	**0.95**	**965**		**0.62**	**0.57**	**1484**		**0.91**	**0.86**	**1173**		**0.05**	**0.00**	**529**
Global Avg.						**3.57**	**3.45**	**97**		**1.35**	**1.23**	**96**		**0.71**	**0.59**	**18**		**1.04**	**0.92**	**258**		**0.38**	**0.27**	**422**		**0.65**	**0.53**	**451**		**0.49**	**0.38**	**176**

Table 5. Detailed results for the third data subset DS3 (Barreto Instances)

Instance	n	m	BKS	BKC	GRASP [2006]				MA|PM [2006]				LRGTS [2007]				GRASP + ELS [2010]				SALRP [2010]				ALNS [2011]				HGTS [2011]			
					Cost	Gap BKS	Gap BKC	CPU time	Cost	Gap BKS	Gap BKC	CPU time	Cost	Gap BKS	Gap BKC	CPU time	Cost	Gap BKS	Gap BKC	CPU time	Cost	Gap BKS	Gap BKC	CPU time	Cost	Gap BKS	Gap BKC	CPU time	Cost	Gap BKS	Gap BKC	CPU time
Christofides69-50x5	50	5	565.6	565.6	599.1	5.92	5.92	3	**565.6**	0.00	0.00	4	586.4	3.68	3.68	3	**565.6**	0.00	0.00	8	**565.6**	0.00	0.00	53	**565.6**	0.00	0.00	73	580.4	2.62	2.62	45
Christofides69-75x10	75	10	844.4	848.9	861.6	2.04	1.50	10	866.1	2.57	2.03	9	863.5	2.26	1.72	10	850.8	0.76	0.22	86	**848.9**	0.53	0.00	127	854.9	1.24	0.71	207	**848.9**	0.53	0.00	94
Christofides69-100x10	100	10	833.4	833.4	861.6	3.38	3.38	26	850.1	2.00	2.00	45	842.9	1.14	1.14	28	**833.4**	0.00	0.00	127	838.3	0.59	0.59	331	835.4	0.24	0.24	403	838.6	0.62	0.62	234
Daskin95-88x8	88	8	355.8	355.8	356.9	0.31	0.31	18	**355.8**	0.00	0.00	34	368.7	3.63	3.63	18	**355.8**	0.00	0.00	130	**355.8**	0.00	0.00	577	**355.8**	0.00	0.00	250	362.0	1.74	1.74	148
Daskin95-150x10	150	10	43919.9	43963.6	44625.2	1.61	1.50	156	44011.7	0.21	0.11	255	44386.3	1.06	0.96	119	**43963.6**	0.10	0.00	1697	45109.4	2.71	2.61	323	44497.2	1.31	1.21	613	44578.9	1.50	1.40	456
Gaskell67-21x5	21	5	424.9	424.9	429.6	1.11	1.11	0	**424.9**	0.00	0.00	0	**424.9**	0.00	0.00	0	**424.9**	0.00	0.00	0	**424.9**	0.00	0.00	18	**424.9**	0.00	0.00	25	**424.9**	0.00	0.00	6
Gaskell67-22x5	22	5	585.1	585.1	**585.1**	0.00	0.00	0	611.8	4.56	4.56	0	587.4	0.39	0.39	0	**585.1**	0.00	0.00	15	**585.1**	0.00	0.00	17	**585.1**	0.00	0.00	21	**585.1**	0.00	0.00	9
Gaskell67-29x5	29	5	512.1	512.1	515.1	0.59	0.59	0	**512.1**	0.00	0.00	1	**512.1**	0.00	0.00	0	**512.1**	0.00	0.00	9	**512.1**	0.00	0.00	24	**512.1**	0.00	0.00	40	**512.1**	0.00	0.00	11
Gaskell67-32x5	32	5	562.2	562.2	571.9	1.73	1.73	1	571.9	1.73	1.73	1	584.6	3.98	3.98	1	**562.2**	0.00	0.00	18	**562.2**	0.00	0.00	27	**562.2**	0.00	0.00	58	**562.2**	0.00	0.00	40
Gaskell67-32x5	32	5	504.3	504.3	**504.3**	0.00	0.00	1	534.7	6.03	6.03	1	504.8	0.10	0.10	1	**504.3**	0.00	0.00	34	**504.3**	0.00	0.00	25	**504.3**	0.00	0.00	55	**504.3**	0.00	0.00	22
Gaskell67-36x5	36	5	460.4	460.4	**460.4**	0.00	0.00	1	485.4	5.43	5.43	1	476.5	3.50	3.50	1	**460.4**	0.00	0.00	0	**460.4**	0.00	0.00	32	**460.4**	0.00	0.00	61	**460.4**	0.00	0.00	39
Min92-27x5	27	5	3062.0	3062.0	**3062.0**	0.00	0.00	0	**3062.0**	0.00	0.00	1	3065.2	0.10	0.10	0	**3062.0**	0.00	0.00	35	**3062.0**	0.00	0.00	23	**3062.0**	0.00	0.00	38	**3062.0**	0.00	0.00	11
Min92-134x8	134	8	5709.0	5709.0	5965.1	4.49	4.49	50	5950.0	4.22	4.22	111	5809.0	1.75	1.75	48	5719.3	0.18	0.18	280	**5709.0**	0.00	0.00	522	5732.6	0.41	0.41	460	5890.6	3.18	3.18	252
Global Avg.						**1.63**	**1.58**	**20**		**2.06**	**2.01**	**36**		**1.66**	**1.61**	**18**		**0.08**	**0.03**	**188**		**0.29**	**0.25**	**161**		**0.25**	**0.20**	**177**		**0.78**	**0.74**	**105**