USING LOCAL SURROGATE INFORMATION IN LAGRANGEAN RELAXATION: AN APPLICATION TO SYMMETRIC TRAVELING SALESMAN PROBLEMS

Luiz Antonio N. Lorena
LAC/INPE
Caixa Postal 515
12.201-970 São José dos Campos - SP - Brazil

Marcelo Gonçalves Narciso
CNPTIA/EMBRAPA - Campinas - SP- Brazil

.

Abstract

The *Traveling Salesman Problem (TSP)* is a classical Combinatorial Optimization problem intensively studied. The *Lagrangean relaxation* was first applied to the *TSP* in 1970. The Lagrangean relaxation limit approximates what is known today as *HK* (Held and Karp) bound, a very good bound (less than 1% from optimal) for a large class of symmetric instances. It became a reference bound for new heuristics, mainly for the very large scale instances, where the use of exact methods is prohibitive. A known problem for the Lagrangean relaxation application is the definition of a convenient step size control in subgradient like methods. Even preserving theoretical convergence properties, a wrong defined control can reflect in performance and increase computational times, a critical point for the large scale instances. We show in this work how to accelerate a classical subgradient method while conserving good approximations to the *HK* bounds. The *surrogate* and Lagrangean relaxation are combined using the local information of the constraints relaxed. It results in a one-dimensional search that corrects the possibly wrong step size and is independent of the used step size control. Comparing with the ordinary subgradient method, and beginning with the same initial multiplier, the computational times are almost twice as fast for medium instances and greatly improved for some large scale *TSPLIB* instances.

Key words: Lagrangean/surrogate relaxation, Traveling Salesman Problem, Subgradient method.

1. Introduction

The *Traveling Salesman Problem (TSP)* is one of the most studied problems in the Combinatorial Optimization literature. Several articles have been published on the subject and it remains today as an interesting and challenging problem. The most common interpretation of the problem seeks the shortest tour for a salesman on a number of cities or clients. Clients must be visited exactly one time and the salesman must return to the home city. For a comprehensive survey of solution methods, applications and related problems see the book of Lawler et al. [27]. Laporte [25] gives another review, including applications on computer wiring, wallpaper cutting, hole punching, job sequencing, dartboard design and crystallography. The problem is well known to be NP-hard [25], justifying the use of heuristics, mainly for large scale problems. Johnson and McGeoch [20] give a recent survey on the use of local search based heuristics.

The Lagrangean relaxation is a well known relaxation technique frequently used to give bound information to combinatorial optimization problems [see for example the survey papers [9, 10] and the books [32, 36]). Held and Karp [17, 18] applied the Lagrangean relaxation to *TSP* in 1970. The relaxation limit approximates what is known today as *HK* (Held and Karp) bound, a very good bound (less than 1% from optimal) for a large class of symmetric instances [21]. Johnson et al. [21] report that exact *HK* bounds have been computed by a special purposed linear programming code, for instances as large as 33810 cities. For even large scale instances, it is applied the subgradient method proposed on the original Held and Karp papers and speeded up by a number of algorithmic tricks [2,16,34,37,38]. Since for large scale instances the optimal solution is not known, the comparison of the heuristic and *HK* bounds is common practice.

Although of simple convergence conditions [8, 33], the convergence of subgradient methods can consume a long computational time for some instances. The subgradient optimization is very sensitive to the values of the initial multipliers and the rules applied for step size control. Efforts were made to have theoretical foundations for these choices [3, 13], but until today the most popular approaches are based on previous empirical experience [19].

Other subgradient methods appeared in literature [4,5,6,23,24,26]. More elaborated, they increase the local computational times computing descent directions [6], or combining subgradients of previous iterations [4,5], or realizing projections onto general convex sets [23,24,26]. Experimental results with some of these methods show an improvement in performance compared to the subgradient method [23,26], but the subgradient method remains the widely used approach in the Lagrangean relaxation context.

Reducing the initial erratic behavior of the subgradient method can result in fast convergence. This can be interesting for large scale problems, even using fast computers. The Lagrangean relaxation is combined with the surrogate relaxation, using the local information (optimization) provided by the relaxed constraints, with the objective of accelerate the subgradient method while conserving the same *HK* bounds. The idea is to introduce a local optimization step at the initial iterations of the subgradient method. The relaxations are applied in sequence. The first relaxation is a surrogate relaxation of the assignment constraints at the *TSP* formulation, followed by a Lagrangean relaxation of the surrogate constraint. A local Lagrangean dual optimization is approximately solved. The process is repeated for a pre-defined number of iterations of the subgradient method. The computational times obtained are almost twice as fast for medium instances and greatly improved for some large scale *TSPLIB* (http://www.crpc.rice.edu/softlib/catalog/tsplib.html) instances.

The combined use of surrogate and Lagrangean relaxation was tested before with success on Set Covering problems [1,28], Generalized Assignment problems [29,31] and some Location problems [35]. Narciso and Lorena [31] coined the name *Lagrangean/surrogate* for this kind of relaxation. Notably is the gain in computer times for large scale instances.

Section 2 presents the *TSP* formulation and the corresponding Lagrangean/surrogate formulation. Section 3 details the subgradient method modified by the local search, and the next section presents computational results for two samples of instances drawn from the *TSPLIB*. We conclude with general comments.

2. *The surrogate information in Lagrangean relaxation*

We initially give an integer linear programming formulation for symmetric *TSPs*. Consider a *TSP* defined on a graph *G = (V,E)* , *V={1,...,n}*, and let the binary variable x_{ij} be equal to 1 if the edge *(i,j)* Î *E* is used in the optimal tour. $C = [c_{ij}]$, where $c_{ij} = c_{ji}$ for all *i,j* Î *V*, is a distance (or cost) matrix associated with *E*. The formulation is

$$Min \sum_{i<j} c_{ij} x_{ij}$$

(P) subject to
$$\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} = 2, \quad k = 1,\dots,n, \tag{1}$$
$$\sum_{i,j \in S} x_{ij} \le |S| - 1, \ \ S \subset V, \ \ 3 \le |S| \le n-3, \tag{2}$$
$$x_{ij} \in \{0,1\}, \ i, j = 1,\dots,n, \ \ i < j. \tag{3}$$

Constraint (1) specify that every vertex has degree 2, constraints (2) are subtour elimination constraints, and (3) the binary conditions. As was point out by Laporte [25] connectivity constraints equivalent to (2) are

$$\sum_{i \in S, j \in V \setminus S} x_{ij} \ge 2, \ \ S \subset V, \ \ 3 \le |S| \le n-3. \tag{4}$$

A well-known relaxation to *(P)* is the length of the 1-spanning tree, obtained by the shortest tree having vertex set V\{1} and two minimal distinct edges at vertex 1. A known formulation is

$$Min \sum_{i<j} c_{ij} x_{ij}$$

(1-T) subject to
$$\sum_{i<j} x_{ij} = n, \tag{5}$$
$$\sum_{j=2}^{n} x_{1j} = 2, \tag{6}$$
$$\sum_{i \in S, j \in V \setminus [S \cup \{1\}]} x_{ij} \ge 1, \ \ S \subset V \setminus \{1\}, \ \ 1 \le |S| \le n-1, \tag{7}$$
$$x_{ij} \in \{0,1\}, \ i, j = 1,\dots,n, \ \ i < j.$$

Constraint (5) is derived taking half the sum of constraints (1), constraint (6) is constraint (1) for k = 1, and constraint (7) is a weaker form of (4) (see [25]).

Problem *(1-T)* is solved in practice by applying a minimum spanning tree algorithm to the graph resulted after the exclusion of vertex 1 and their end point edges [25]. The vertex 1 is then included at the resulting tree, adding the two minimum costs edges that connects vertex 1 to the tree.

Held and Karp reinforced the *(1-T)* bound using *Lagrangean relaxation*. Considering the multipliers $l_k,\ k \in V$, constraints (1) are relaxed in the objective function obtaining the following Lagrangean function

$$L(\,l\,) = Min_x \left\{ \sum_{i<j} c_{ij} x_{ij} + \sum_{k \in V} l_k \left(\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} - 2 \right) \right\} ,$$

where x is a feasible solution to *(1-T)*. The Lagrangean bound is improved by searching the solution of the Lagrangean dual problem $D(\,l) = Max_{\,l}\{L(\,l\,)\}$.

The surrogate duality theory is an old matter, that was not so intensively explored like the Lagrangean counterpart (see the papers [7, 11, 12, 14, 22] and the book [32] for a formal view of the subject). We explore here the simple relationship between the two relaxations, recalling that Lagrangean multipliers can also be considered as surrogate multipliers, and making profit of the local optimization induced by a new local Lagrangean relaxation.

The multipliers $l_k,\ k \in V$, can be seen as surrogate multipliers, and constraint $\sum_{k \in V} l_k \left(\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} - 2 \right) = 0$, as a surrogate constraint included in problem *(1-T)*. Using a one-dimensional multiplier *t Î R*, and relaxing this surrogate constraint in the Lagrangean way, we obtain the surrogate version of the Lagrangean function (named *Lagrangean/surrogate* in [31])

$$L_t(\,l\,) = Min_x \left\{ \sum_{i<j} c_{ij} x_{ij} + \sum_{k \in V} t.\ l_k \left(\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} - 2 \right) \right\} ,$$

where x is a feasible solution to *(1-T)*.

In our notation, $L(l) = L_1(l)$. For a given l, a local dual can be identified here, as $D_t(l) = Max_t\{L_t(l)\}$. It is interesting to note that for $t = 1$ the local optimization induced by the surrogate constraint is not considered. The same condition is observed for each fixed value of t. It is also immediate that for the same l, $v\,[D_t(l)]$ ³ $v[L(l)]$, i.e., the local dual gives an improved bound to the usual Lagrangean relaxation (v[(.)] is an optimal value for problem (.)).

It is well known that the Lagrangean function is concave and piecewise linear [9]. An exact solution to $D_t(l)$ may be obtained by a search over different values of t (see Minoux [30] and Handler and Zang [15]). However, in general, we have an interval of values t_0 £ t £ t_1 (with $t_0 = 1$ or $t_1 = 1$) which also produces improved bounds to the usual Lagrangean ones (see Figure 1, for the case $t_1 = 1$).

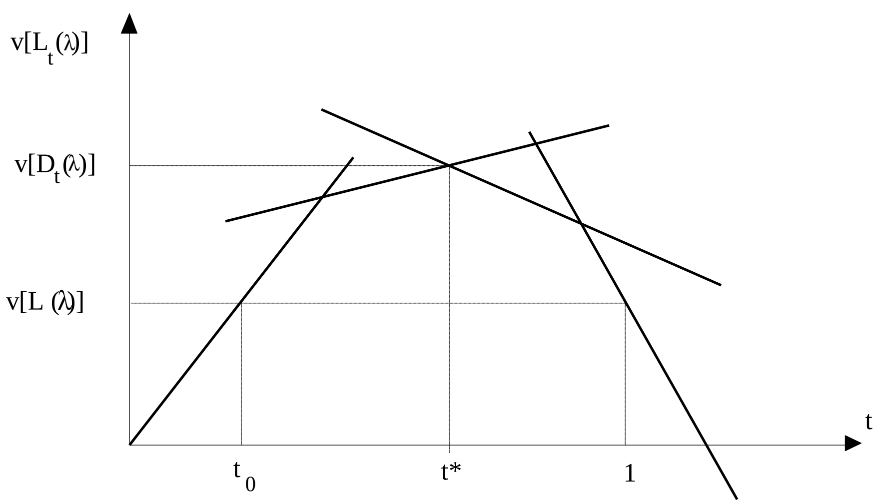

Figure 1: Lagrangean/surrogate bounds.

So, in order to obtain an improved bound to the usual Lagrangean relaxation it is not necessary to find the best value t^*, being enough to find a value T, such as t_0 £ T £ t_1. The following inequalities are valid, $v(P)$ ³ $v[D(l)]$ ³ $v[D_t(l)]$ ³ $v[L_T(l)]$ ³

v[L(l)]. The Lagrangean/surrogate bound is a better local limit than the Lagrangean bound, but the overall dual optimization produces the same theoretical bounds (for either Lagrangean alone or Lagrangean/surrogate [31]).

3. The subgradient method

The subgradient method is employed to solve problem *D(l)*, giving an approximated *HK* bound for problem *(P)*. We propose here to use the traditional subgradient method, with the step size corrections provided by Held and Karp [18], without any modification or improvement. That decision will respond the question if the original HK step was a good one. Observing the literature for other suggestions on step size corrections and/or new step sizes, it become evident the necessity of such modifications [3, 5, 16, 21, 34, 37, 38].

Beginning with the same initial multiplier l^0 , a different sequence of relaxation bounds is obtained for the usual Lagrangean (*t* fixed in 1 at all iterations) and the Lagrangean/surrogate (*t* is calculated for a number of iterations and then fixed). The multiplier updates observe the following formula

$$l^{i+1} = l^i + b\,[v_f - v(L_t(l^i)]g_t^{l^i} / \left\|g_t^{l^i}\right\|^2,\ 0 \le b \le 2 \qquad (8)$$

(where v_f is the value of a feasible solution to (P)).

It is easy to see different sequences observing that the subgradients are distinct, $g_t^l \neq g_1^l$ (in general). The parameter b follows the Held and Karp [19] suggestion, that makes $0 \le b \le 2$, beginning with $b = 2$. If after 20 iterations $v[L_t(l)]$ not increases, b is updated to $b = b/2$.

The value T suggested in *figure 1* for t , can be obtained by a simple one-dimensional search. Beginning with an initial t , many types of search can be employed here, but the ideal will be that one making the smallest number of $v[L_t(l)]$ evaluations to reach the interval t_0 £ T £ t_1 . The following one-dimensional search was used. The value of T is increased while the slope of the Lagrangean/surrogate function is positive (or for a pre-fixed number of iterations).

---

t-search

Given

λ ; *(current Lagrangean multiplier)*

increment = 1.5;

k_max = 5; *(maximum number of iterations)*

t_0 := -¥ ; *(lower bound for the best t)*

T := increment; *(initial Lagrangean/surrogate multiplier)*

t_1 := ¥ ; *(upper bound for the best t)*

v := -¥ ;* *(best Lagrangean/surrogate bound)*

k := 0; *(number of iterations)*

While *k £ k_max* **do**

    k := k + 1;

    solve $L_T(l)$

    _If_ $v[L_T(l)] > v^*$ **_then_** $v^* := v[L_T(l)]$;

    _If_ $\sum_{k \in V} l_k \left(\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} - 2 \right) < 0$ **_then_**

        t_1 = T;

        T = T - increment;

        _If_ *t_0 ¹ -¥ (t_0 was already determined)* **_then_**

            increment = (t_1 – t_0)/2 ;

            T = T + increment;

        _End_If_

        Stop;

    else

        t_0 = T;

        *increment = increment*2;*

        T = T + increment;

    _End_If_

_End_while_

---

4. Computational tests

A sample of symmetric instances was initially selected from the *TSPLIB* (http://www.crpc.rice.edu/softlib/catalog/tsplib.html) to conduct a computational comparison between the application of the usual Lagrangean relaxation (multiplier t is fixed to 1 at each iteration of the subgradient method) and the Lagrangean/surrogate (which explores the one-dimensional search for t at some of the initial iterations of the subgradient method).

This initial set of instances is composed of the problems known as: *uly**16**m; uly**22**m; att**48**; berlin**52**; kroA**100**; tsp**225**; pcb**442**; pr**1002**; d**1291**, rl**1304**; nrw**1379**; d**1655**; vm**1748**; rl**1889** and u**2152*** (refereed herein as ***16**, **22**, **48**, **52**, **100**, **225**, **442**, **1002**, **1291**, **1304**, **1379**, **1655**, **1748**, **1889 and 2152**).*

Table 1 presents the results for the usual Lagrangean relaxation, while *table 2* presents the results for the Lagrangean/surrogate counterpart. The algorithms are coded in *C* and run on a *SUN ULTRA1 127 Mhz*. The columns in tables are composed of:

Prob. ® *problem instance,*
n_iter ® *number of iterations (limited to 3000),*
time ® *total computer time,*
gap ® *(optimal solution – relaxation)/optimal solution,*
10%, 5%, 4%, 3%, 2%, 1%, 0.5%, 0.4%, 0.3%, 0.2%, 0.1% ®
elapsed time to gap be equal to a *%* , a *Î {0.1, 0.2, 0.3, 0.4, 0.5, 1, 2, 3, 4, 5, 10}.*

The experiments were conducted to compare lower bounds, and then the known optimal solution value was used for v_f on the multiplier updating formula (8). The same initial multiplier was used at the subgradient method (l^0 = (1,1,…,1)) for both relaxation. The stop conditions are the iteration limit (3000), or for a small β (< 0.005), or if $v_f - v[L_t(l)] < 1$. The gap percentages reflect the behavior of the subgradient method without the effect of stopping tests, and the smaller one is used to compare the relaxations.

Comparing the results in *tables 1 and 2,* we can see that the Lagrangean/surrogate reaches tighter gaps than the Lagrangean ones, using only part of the time required by the Lagrangean, and the same gaps with a notably saving of time for the large scale problems.

To better compare, *table 3* shows for each problem the Lagrangean and Lagrangean/surrogate (enclosed in brackets) results: *best gap* (%), *elapsed time* to reach the Lagrangean best gap, and finally the percentage of time expended by the Lagrangean/surrogate to reach the same gap attained by the Lagrangean. The Lagrangean/surrogate was able to reach 6 tighter bounds, all after the problem size of ***1002***. Observing the last column we can see, for example, that it reached the better Lagrangean bound using only 2.6 % of time on problem ***1889*** , 2.8% on problem *1002*, 3.7% on problem ***1748***, and 5.3% on problem ***1304***. The economy of time was not representative for two small instances (***52*** and ***225***).

Problem	*Best gap (%)*		*Times (sec.)*		*Time (%)*
16	0.1	(0.1)	2.	(1.03)	51
22	0.1	(0.1)	9.1	(4.6)	51
48	0.3	(0.3)	19.	(8.)	42
52	0.3	(0.3)	5.	(5.)	100
100	2.	(2.)	27.	(14.)	51
225	4.	(4.)	495.	(392.)	92
442	1.	(1.)	4054.	(997.)	24
1002	4.	(2.)	36714.	(1054.8)	2.8
1291	3.	(3.)	13431.	(3230.)	24
1304	5.	(2.)	28094.3	(1511.)	5.3
1379	2.	(2.)	9465.7	(3147.)	33
1655	3.	(2.)	29368.	(3029.)	10
1748	5.	(2.)	48413.	(1802.)	3.7
1889	5.	(2.)	87568.	(2275.4)	2.6
2152	2.	(1.)	31334.	(3648.)	11.6

Table 3: Comparison: Lagrangean versus Lagrangean/surrogate – first set of instances

It appears from *table 3* that the Lagrangean/surrogate do not improve the Lagrangean times for small instances (< 1000 cities), and greatly improves it for the large instances (> 1000 cities). We have then proceeded the computational tests to reinforce this observation.

The second set of instances is composed of the problems known as: *st**70**; bier**127**; gr**137**; ch**150**; gr**202**, a**280**; lin**318**; gr**431**; att**532**; rat**575**; rat**783**; u**2319**; pr**2392** and pcb**3038*** (refereed herein as ***70**, **127**, **150**, **202**, **280**, **318**, **431**, **532**, **575**, **783**, **2319**, **2392 and 3038**).*

Table 4 presents results on a similar way to that one presented in *table3*. The Lagrangean/surrogate was able to reach 5 tighter bounds. For problems ***280*** and ***575***, the Lagrangean times are better than the Lagrangean/surrogate ones. But, for example, on problems ***202*** and ***431***, the Lagrangean/surrogate employed only 7% and 6.62% of the time needed by the Lagrangean to reach their best bound. For the large scale instances, it is confirmed the great improvement in times, particularly on instance 2319, where the Lagrangean/surrogate used only 2.69% of the time to reach the Lagrangean best bound.

Problem	*Best gap (%)*		*Times (sec.)*		*Time (%)*
70	4.	(4.)	4.	(4.)	100.
127	10.	(1.)	243.	(12.)	4.9
150	2.	(2.)	26.	(18.)	69.2
202	3.	(0.3)	1495.	(105.)	7.
280	2.	(2.)	22.	(80)	363
318	2.	(1.)	679.	(62.)	9.13
431	10.	(2.)	1450.	(96.)	6.62
532	2.	(2.)	1799.	(580.)	32.2
575	4.	(4.)	106.	(464.)	437.
783	10.	(10.)	122.	(112.)	91.8
2319	10.	(1.)	92819.	(2503.)	2.69
2392	4.	(2.)	63401.	(4161.)	6.56
3038	2.	(2.)	80661.	(4294.)	5.32

Table 4: Comparison: Lagrangean (Lagrangean/surrogate) – second set of instances

The Lagrangean/surrogate relaxation requires the application of the *t-search* algorithm for a number of initial iterations. The criterion used was to fix the T value if it repeats for 5 consecutive iterations. For almost all the cases, the T value was fixed on 46.5, and in some cases on 22.5. It is a direct consequence of the one-dimensional search used. It is also observed that, in general, T was fixed after the *5* initial iterations, that has the same effect that if it was fixed at the *first* iteration.

The best required t can result in a value that is very large than the usual Lagrangean t (= 1), and the local search produced relevant effects for these instances, reflecting on the behavior of the relaxation sequences. It can be better observed on *figure 2*. Three instances are used, the *48*, *442* and *1002*. The local search effects can be seen on the initial perturbed sequences for the Lagrangean/surrogate case. The Lagrangean sequences were very stable, but increase at small rates (slopes), mainly for the *1002* instance, where the Lagrangean/surrogate employed only 2.8% of the time needed by the Lagrangean to reach their best bound.

One conclusion based on this *TSPLIB* sample of instances is that $t = 1$ is not the best multiplier for almost all the instances tested, justifying the search for better Lagrangean performance on *TSP* [3, 5, 16, 21, 34, 37, 38].

5. Conclusions

We investigated in this paper the effects of local search on Lagrangean relaxation applied to symmetric TSP. The local search was simply justified considering the Lagrangean multipliers as surrogate multipliers, affected by a local one-dimensional Lagrangean dual. The local search can be a straight one, giving in few steps a better one-dimensional multiplier than the usual Lagrangean multiplier (fixed in one).

The name *Lagrangean/surrogate*, coined at Narciso and Lorena [31] paper can be used to reflect the local search use on Lagrangean relaxation (see the related works [1, 28. 29. 35]). For two samples of instances drawn from the *TSPLI*B, it produced tight gaps compared with the usual Lagrangean ones, and for large scale instances, considerably small times for the same gaps.

We hope that the Lagrangean/surrogate approach can be useful for even large scale TSP instances, considering the importance of *HK* bounds for heuristic performance comparison [20, 21]. It is also important to note that the refereed approach is independent of the step size and subgradient direction used (if the convergence conditions were observed).

Acknowledgments:

The authors acknowledge the comments and suggestions made by two anonymous referees.

The first author acknowledges Fundação para o Amparo a Pesquisa no Estado de São Paulo – FAPESP (proc. 96/04585-6) and Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (proc. 350034/91-5) for partial financial research support.

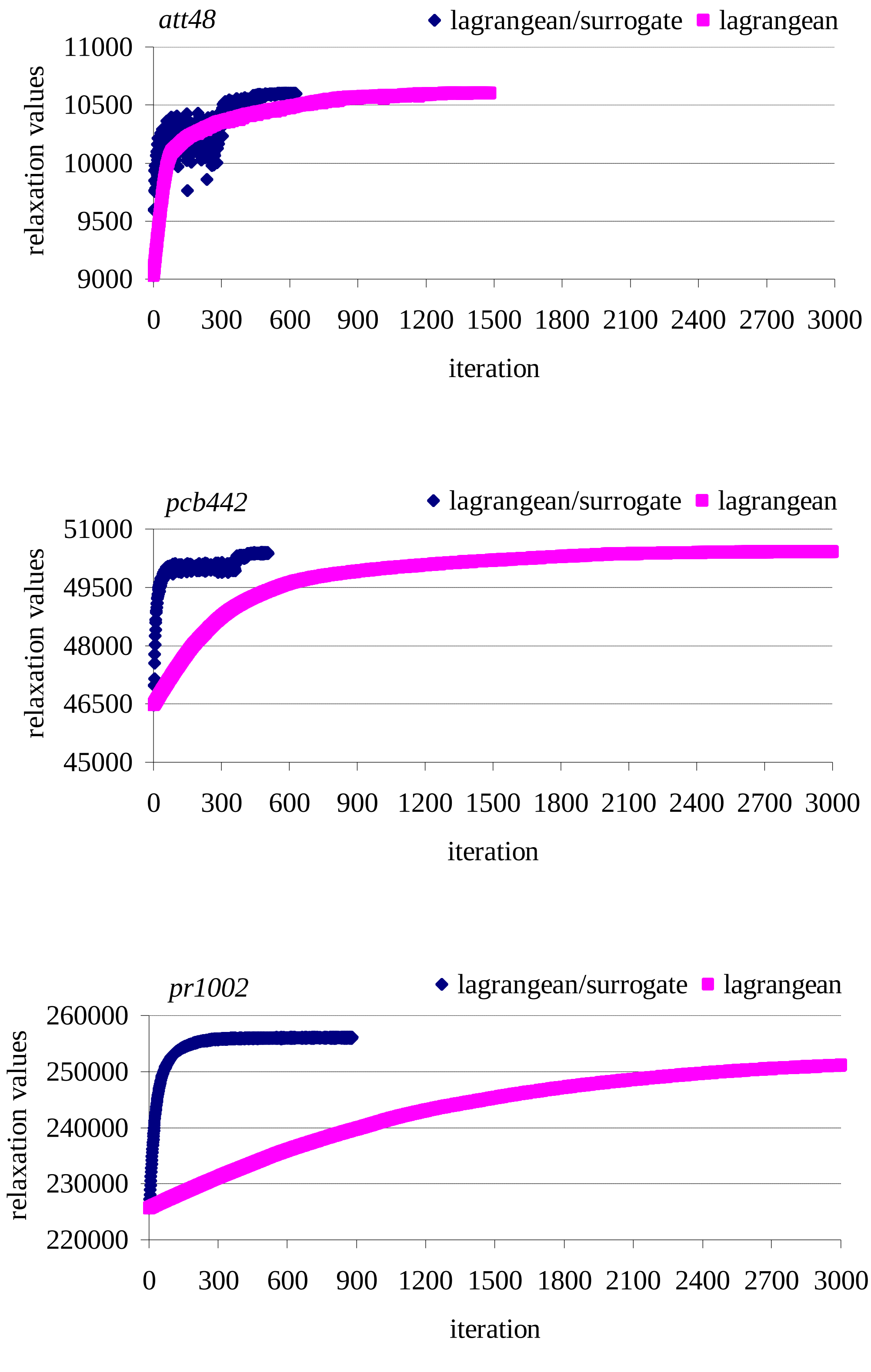

Figure 2: Lagrangean/surrogate versus Lagrangean – *att48, pcb442 and pr1002*

Prob.	*n_iter*	*time*	*gap*	*10%*	*5%*	*4%*	*3%*	*2%*	*1%*	*0.5%*	*0.4%*	*0.3%*	*0.2%*	*0.1%*
16	289	2.	0.000230	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.	2.
22	487	9.38	0.000123	2.9	4.1	4.3	4.6	4.8	6.3	7.7	7.9	8.4	8.7	9.1
48	1431	23.	0.002355	1.	1.	2.	4.	7.	11.	15.	17.	19.	--	--
52	282	6.	0.002132	1.	2.	2.	2.	3.	4.	5.	5.	5.	--	--
100	714	52.	0.018157	0.	6.	9.	15.	27.	--	--	--	--	--	--
225	950	675.	0.039181	2.	67.	495.	--	--	--	--	--	--	--	--
442	3000	6469.	0.007115	2.	527.	754.	1119.	1810.	4054.	--	--	--	--	--
1002	3000	52420.	0.030597	6853.	27475.	36714.	--	--	--	--	--	--	--	--
1291	3000	56066.	0.023842	14.	420.	2376.	13431.	--	--	--	--	--	--	--
1304	3000	42816.	0.040637	4234.9	28094.3	--	--	--	--	--	--	--	--	--
1379	2986	38018.	0.015077	9.9	1918.6	2826.1	4412.8	9465.7	--	--	--	--	--	--
1655	3000	128326.	0.02204	40.1	6405.2	12701.9	29368.	--	--	--	--	--	--	--
1748	3000	64421.	0.040159	9739.	48413.	--	--	--	--	--	--	--	--	--
1889	3000	87629.	0.049982	10786.	87568.	--	--	--	--	--	--	--	--	--
2152	3000	99230.	0.012201	21.4	21.4	2900.1	11413.	31334.	--	--	--	--	--	--

Table 1 : TSPLIB instances – Lagrangean results.

Prob.	*n_iter*	*time*	*gap*	*10%*	*5%*	*4%*	*3%*	*2%*	*1%*	*0.5%*	*0.4%*	*0.3%*	*0.2%*	*0.1%*
16	264	1.1	0.000233	0.1	0.1	0.2	0.2	0.8	0.9	1.	1.	1.	1.	1.03
22	372	7.	0.000096	1.0	1.1	1.2	1.2	1.4	1.7	2.	2.3	2.5	3.9	4.6
48	521	8.	0.002988	0.15	1.	1.	1.	3.	6.	7.	7.	8.	--	--
52	309	6.5	0.002121	0.33	1.	1.	1.	4.	5.	5.	5.	5.	--	--
100	373	28.	0.021871	0.56	3.	3.	4.	14.	--	--	--	--	--	--
225	882	652.	0.039154	2.28	83.	392.	--	--	--	--	--	--	--	--
442	506	997.	0.009726	1.95	82.	92.	110.	152.	997.	--	--	--	--	--
1002	905	17856.	0.011068	455.5	869.5	1054.8	1428.4	2514.5	--	--	--	--	--	--
1291	618	15384.	0.021880	14.	291.	736.	3230.	--	--	--	--	--	--	--
1304	1057	15033.	0.018360	626.	1511.	2111.	3678.	9060.	--	--	--	--	--	--
1379	962	12249.	0.014109	9.15	315.	351.	440.	3147.	--	--	--	--	--	--
1655	3000	128325.	0.019877	40.1	1251.	1600.	3029.	42846.	--	--	--	--	--	--
1748	3000	64419.	0.014932	785.	1802.	2285.	3098.	6716.	--	--	--	--	--	--
1889	3000	87643.	0.017504	675.8	2275.4	3785.9	9080.7	37393.	--	--	--	--	--	--
2152	3000	99222.	0.009182	25.7	25.7	1106.	1805.	3648.	21829.	--	--	--	--	--

Table 2 : TSPLIB instances – Lagrangean/surrogate results

References

[1] Almiñana, Marcos e Pastor, T. J. "An adaptation of SH heuristic to the location set covering problem". European Journal of Operational Research, 100, 586-593, 1997.

[2] Allen, E., Helgason, R., Kennington, et al. "A generalization of Poliak's convergence results for subgradient optimization". Mathematical Programming, 37, 309-317, 1987.

[3] Bazaraa, M. S., Sherali, H. D. "On the choice of step size in subgradient optimization". European Journal of Operational Research, 7, 380-388, 1981.

[4] Brännlund, U., "A Generalized subgradient method with relaxation step", *Mathematical Programming* 71 (1995) 207-219.

[5] Camerini, P. ; Fratta, L. and Maffioli F., "On improving relaxation methods by modified gradient techniques" *Mathematical Programming Study,* 3, (1975) 26-34.

[6] Correa, R. and Lemaréchal, C, "Convergence of some algorithms for convex minimization", *Mathematical Programming* 62 (1993) 261-275.

[7] Dyer, M. E. "Calculating surrogate constraints". Mathematical Programming, 19, 255-278, 1980.

[8] Ermol'ev, Y. M. "Methods for solving nonlinear extremal problems". Cybernetics, 16/1, 1-14, 1966.

[9] Fisher, M. L. "The lagrangian relaxation method of solving integer programming problems". Management Science, 27, 1-18, 1981.

[10] Geoffrion, A. " Lagrangean relaxation and its uses in integer programming". Mathematical Programming Study, 2, 82-114, 1974.

[11] Glover, F. "Surrogate constraints". Operations Research, 16(4):741-749, 1968.

[12] Glover, F. "Surrogate Constraints Duality in Mathematical Programming". Operations Research, 23, 434-451, 1975.

[13] Goffin, J. L. "On convergence rates of subgradient optimization methods", Mathematical Programming, 13, 329-347, 1977.

[14] Greenberg, H. J., Pierskalla, W. P. "Surrogate Mathematical Programming". Operations Research, 18, 924-939, 1970.

[15] Handler, G., Zang, I. "A dual algorithm for the constrained shortest path problem". Networks, 10, 193-310, 1980.

[16] Helbig-Hansen, K. H., Krarup, J. "Improvements of the Held-Karp algorithm for the symmetric traveling salesman problem". Mathematical Programming, 7, 87-96, 1974.

[17] Held, M., Karp, R. M. " The Traveling salesman problem and minimum spanning trees". Operations Research, 18,1138-1162, (1970).

[18] Held, M., Karp, R. M. " The Traveling salesman problem and minimum spanning trees: Part II". Mathematical Programming 1, 6-25, (1971).

[19] Held. M., Wolfe, P., Crowder, H. P. "Validation of subgradient optimization", Mathematical Programming, 6, 62-88, 1974.

[20] Jonhson, D.S., McGeoch, L.A. "The traveling salesman problem: a case study in local optimization". In *Local search in Combinatorial optimization,* E. H. L. Aarts and J. K. Lenstra (eds.), John Wiley & Sons, New York, 1997.

[21] Jonhson, D.S., McGeoch, L.A., Rothberg, E. E. " Asymptotic Experimental Analysis for the Held-Karp Traveling Salesman Bound. Proceedings of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms, 341-350, 1996.

[22] Karwan, M. H., Rardin, R. L. "Some relationships between lagrangian and surrogate duality in integer programming". Mathematical Programming, 17, 320-334, 1979.

[23] Kim, S. and Um, B. S., "Polyak's subgradient method with simplified projection for nondifferentiable optimization with linear constraints", *Optimization* 20 (1989) 451-456.

[24] Kim, S. and Um, B. S., " An improved subgradient method for constrained nondifferentiable optimization", *Operations Research Letters* 14 (1993) 61-64.

[25] Laporte, G. "The traveling salesman problem: an overview of exact and approximate algorithms". European Journal of Operational Research, 59, 231-247, 1992.

[26] Larsson, T. ; Patriksson, M. and Strömberg, A-B., "Conditional subgradient optimization - theory and applications", *European Journal of Operational Research* 88, 382-403, 1996.

[27] Lawler, E. L. ; Lenstra, J. K. ; Rinnooy Kan, A. H. G. and Shmoys, D. B., “The traveling salesman problem, John Wiley and Sons, Chichester, 1985.

[28] Lorena, L. A. N., Lopes, F. B. " A surrogate heuristic for set covering problems". European Journal of Operational Research. 79(1), 138-150, 1994.

[29] Lorena, L. A. N., Narciso, M. G. "Relaxation heuristics for a generalized assignment problem". European Journal of Operational Research, 91(1), 600-610, 1996.

[30] Minoux, M.. "Plus courts chemins avec constraints: Algorithmes et applications", Annals of Telecommunications, 30, 383-394, 1975.

[31] Narciso, M. G., Lorena, L. A.N. "Lagrangean/surrogate relaxation for generalized assignment problem". European Journal of Operational Research. To appear(1999)

[32] Parker, R. G., Rardin, L. R. "Discrete Optimization". Academic Press, INC, London, 1988.

[33] Poljak, B. T. "Minimization ofunsmooth functionals". USSR Computational Mathematics and Mathematical Physics 9, 14-29, 1969.

[34] Reinelt. G. “The traveling salesman problem: computational solutions for TSP applications”. Lecture Notes in Computer Science 840, Springer Verlag, Berlin, 1994.

[35] Senne, L. F.; Lorena L. A. N. “Lagrangean/Surrogate Heuristics for p-Median Problems”. In Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research, M. Laguna and J. L. Gonzalez-Velarde (Eds.), Kluwer Academic Publishers, pp. 115-130, 2000.

[36] Shapiro, J. F. "Generalizedlagrange multipliers in integer programming". Operations Research, 19, 68-76, 1971.

[37] Volgenant, T., Jonker, R. “ A branch and bound algorithm for the symmetric traveling salesman problem based on the 1-tree relaxation”. European Journal of Operational Research, 9, 83-89, 1982.

[38] Valenzuela, C. L., Jones, A J. “Estimating Held-Karp lower bond for the geometric TSP”, 1995.