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

Luiz Antonio N. Lorena
INPE – Instituto Nacional de Pesquisas Espaciais
LAC – Laboratório Associado de Computação e Matemática Aplicada
Caixa Postal 515
12.201-970 São José dos Campos - SP – Brazil
Tel. +55-012-3456553
FAX +55-012-3456375
lorena@lac.inpe.br

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

.

Abstract

The *Traveling Salesman Problem (TSP)* is a classical Combinatorial Optimization problem, which has been 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 affect the performance and increase computational times. 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 relaxed constraints. 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, 36] and the book [32]). Held and Karp [17, 18] applied the Lagrangean relaxation to *TSP* in 1970. The bound that their relaxation gives is known today as *HK* (Held and Karp) bound. This bound performs very well (less than 1% from optimal) on 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.

In spite of the 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 initial values of the multipliers and the rules applied to control the step size. Efforts have done to develop theoretical foundations for these

choices [3, 13]. Unfortunately the most popular approaches are based on previous empirical experience so far [19].

There are many other subgradient methods in the literature [4,5,6,23,24,26]. 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* instances [34].

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.

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 $(i,j) \in E$ is used in the optimal tour and 0 otherwise. $C = [c_{ij}]$, where $c_{ij} = c_{ji}$ for all $i,j \in V$, is a distance (or cost) matrix associated with the edges. Then the formulation of the *TSP* is

$$(P): \quad Min \sum_{i<j} c_{ij} x_{ij}$$

$$\text{subject to} \quad \sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} = 2, \quad k = 1,\dots,n, \tag{1}$$

$$\sum_{i,j\in S} x_{ij} \le |S| - 1, \quad S \subset V, \quad 3 \le |S| \le n-3, \tag{2}$$

$$x_{ij} \in \{0,1\}, \; i, j = 1,\dots,n, \quad i < j. \tag{3}$$

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

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

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

$$(1\text{-}T): \quad Min \sum_{i<j} c_{ij} x_{ij}$$

$$\text{subject to} \quad \sum_{i<j} x_{ij} = n, \tag{5}$$

$$\sum_{j=2}^{n} x_{1j} = 2, \tag{6}$$

$$\sum_{i\in S,\, j\in V\setminus[S\cup\{1\}]} x_{ij} \geq 1, \quad S\subset V\setminus\{1\},\ \ 1\leq |S|\leq n-1, \qquad (7)$$
$$x_{ij}\in\{0,1\},\ i,j=1,\ldots,n,\ \ i<j.$$

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 the incident edges [25]. Vertex 1 is then attached to the resulting tree by 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 λ_k, $k\in V$, constraints (1) are relaxed in the objective function obtaining the following Lagrangean function $L(\lambda) = Min_x\left\{\sum_{i<j} c_{ij}x_{ij} + \sum_{k\in V}\lambda_k\left(\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} - 2\right)\right\}$, where x is a feasible solution to *(1-T)*. The Lagrangean bound is improved by searching the solution of the Lagrangean dual problem $D(\lambda) = Max_\lambda\{L(\lambda)\}$.

Although surrogate duality theory was introduced a while ago it has not been extensively studied like its Lagrangean counterpart (see the papers [7, 11, 12, 14, 22] and the book [32] for a formal representation of the subject). We explore here the simple relationship between the two relaxations by recalling that Lagrangean multipliers can also be considered as surrogate multipliers and benefiting from the local optimization based on a new local Lagrangean relaxation.

The multipliers λ_k, $k\in V$, can be seen as surrogate multipliers, and constraint $\sum_{k\in V}\lambda_k\left(\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} - 2\right) = 0$ as a surrogate constraint included in problem *(1-T)*. Using a one-dimensional real multiplier t, and relaxing this surrogate constraint in the

Lagrangean relaxation form, we obtain the surrogate version of the Lagrangean function (named *Lagrangean/surrogate* in [31])

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

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

In our notation, $L(\lambda) = L_1(\lambda)$. For a given λ, a local dual can be identified as $D_t(\lambda) = Max_t\{L_t(\lambda)\}$. It is interesting to note that for $t = 1$ the local optimization, which is based on 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 λ, $v[D_t(\lambda)] \geq v[L(\lambda)]$. In other words, local dual gives an improved bound on the optimal value of the Lagrangean relaxation. Note that 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 $D_t(\lambda)$ 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 $t_0 \leq t \leq t_1$ (with $t_0 = 1$ or $t_1 = 1$) which also produces improved bounds for the usual Lagrangean relaxation. This is illustrated for the $t_1 = 1$ in Figure 1.

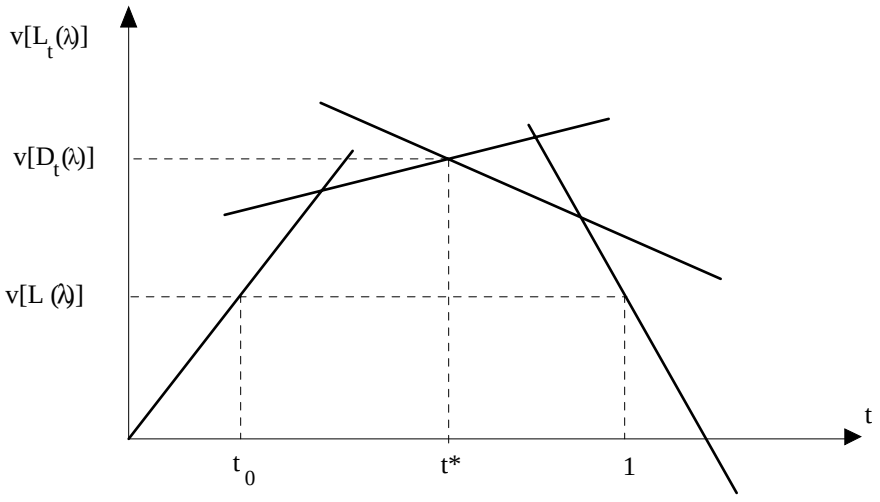

Figure 1: Lagrangean/surrogate bounds.

So, in order to obtain an improved bound for the usual Lagrangean relaxation, it is not necessary to find the best value t^*; a value T such as $t_0 \leq T \leq t_1$ is good enough. The following inequalities are valid, $v(P) \geq v[D(\lambda)] \geq v[D_t(\lambda)] \geq v[L_T(\lambda)] \geq v[L(\lambda)]$. 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(λ)*, 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. This decision will show whether the original HK step is a good one. However many works in the literature suggest better strategies for step size selection [3, 5, 16, 21, 34, 37, 38].

Beginning with the same initial multiplier λ^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 are realized according the following formula

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

Where v_f is the value of a feasible solution to (P), and g_t^{λ} a subgradient.

It is easy to observe different sequences since the subgradients are distinct, i. e., $g_t^{\lambda} \neq g_1^{\lambda}$ (in general). The parameter β follows the Held and Karp [19] suggestion. Namely, it is selected such that $0 \leq \beta \leq 2$, and initialized with $\beta = 2$. If after 20 iterations $v[L_t(\lambda)]$ does not increases, β is updated with $\beta = \beta / 2$.

The value T, which is 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 the one of providing the smallest number of $v[L_t(\lambda)]$ evaluations to reach the interval $t_0 \leq T \leq t_1$. The following one-dimensional search was used. The value of T is increased as long as the slope of the Lagrangean/surrogate function is positive (or for a pre-fixed number of iterations).

Algorithm: ***t-search***

Given

λ ;	*(current Lagrangean multiplier)*
increment = 1.5;	
k_max = 5;	*(maximum number of iterations)*
$t_0 := -\infty$;	*(lower bound for the best t)*
T := increment;	*(initial Lagrangean/surrogate multiplier)*
$t_1 := \infty$;	*(upper bound for the best t)*
$v^* := -\infty$;	*(best Lagrangean/surrogate bound)*
k := 0;	*(number of iterations)*

While $k \leq k\_max$ ***do***

$k := k + 1$;

solve $L_T(\lambda)$

If $v[L_T(\lambda)] > v^*$ ***then*** $v^* := v[L_T(\lambda)]$;

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

$t_1 = T$;

$T = T$ - *increment;*

If $t_0 \neq -\infty$ *(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* [34] 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***. They are refereed 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 relaxation. The algorithms are coded in C and run on a *SUN ULTRA1 127 Mhz* workstation working in the operating system environment. In the column headings of the tables: *prob. , n_iter, time,* and *gap* respectively stand for *problem instance, number of iterations (limited to 3000), total computer times*, and *(optimal solution – relaxation)/optimal solution.* As for the percentages *10%, 5%, 4%, 3%, 2%, 1%, 0.5%, 0.4%, 0.3%, 0.2%, 0.1%* , their related columns give the elapsed times to obtain a gap of $\alpha\%$, with $\alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5, 1, 2, 3, 4, 5, 10\}$.

The experiments were conducted to compare lower bounds, and then the known optimal solution value was used for v_f on the multiplier update formula (8). The same initial multiplier was used in the subgradient method (λ^0 = (1,1,…,1)) for both relaxation. The algorithm stops either when the iteration goes beyond the limit (say 3000), or β becomes small enough (say less than 0.005), or $v_f - v[L_t(\lambda)] < 1$. The gap percentages reflect the behavior of the subgradient method without the effect of stopping conditions, and the smallest one is used to compare the relaxations.

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

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. The Lagrangean/surrogate was able to reach 6 tighter bounds. They are for problems with 1002 cities or larger. Observing the last column we can see, for example, that it reached better bound in only 2.6 % of time used by the Lagrangean relaxation on problem ***1889*** **.** These percentages are 2.8% on problem *1002*, 3.7% on problem ***1748***, and 5.3% on problem ***1304***.

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 does not improve very much the Lagrangean times for small instances (< 1000 cities), and greatly improves it for the large

instances (> 1000 cities). We have then directed the computational tests to emphasis 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***, which we refer as ***70**, **127**, **150**, **202**, **280**, **318**, **431**, **532**, **575**, **783**, **2319, 2392 and 3038***. The results are summarized in *Table 4*. 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 relaxation used only 7% and 6.62% of the time needed by the Lagrangean relaxation to reach their best bound. The improvement in times is very good for large scale instances, in particular on instance 2319, where the Lagrangean/surrogate relaxation used only 2.69% of the time used by the Lagrangean relaxation.

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 to 46.5, and in

some cases to *22.5*. It is a direct consequence of the one-dimensional search used. It is also observed that T was fixed after the five first iterations and has the same effect as if it was fixed at the first.

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. This can be observed better on *figure 2*. Three instances are used, the *48*, *442* and *1002*. The effect of local search 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 uses only 2.8% of the time needed by the Lagrangean relaxation to reach the 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

In this paper we investigated 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 a paper by Narciso and Lorena [31] 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 tighter gaps compared with the ones obtained by using usual Lagrangean relaxation. Besides, considerable time savings occur especially for large scale instances.

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 this 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 Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (proc. 300837/89-5) and Fundação para o Amparo a Pesquisa no Estado de S. Paulo - FAPESP (proc. 99/06954-7) for partial financial research support.

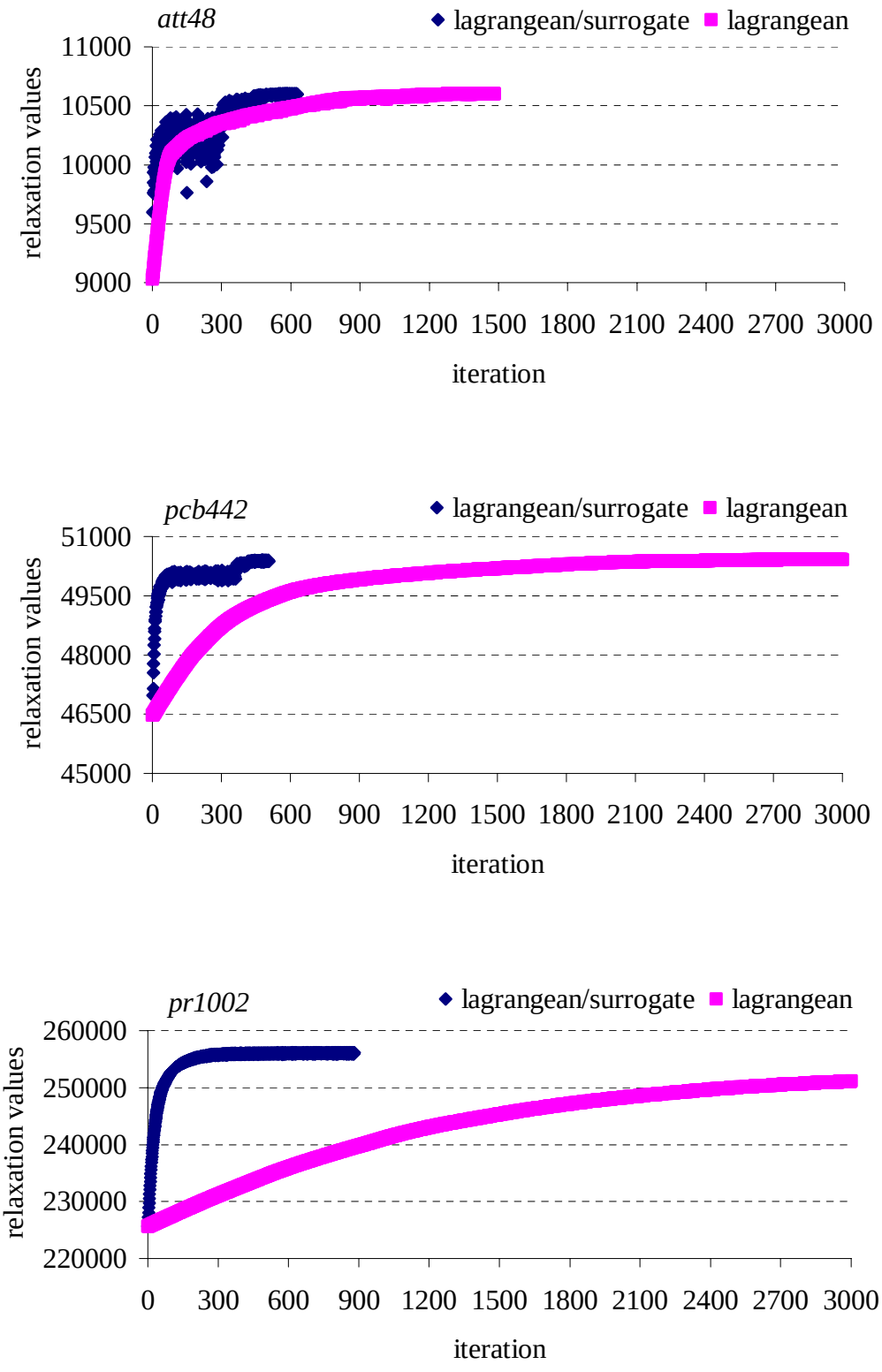

Figure 2: Lagrangean/surrogate versus Lagrangean – *att48, pcb442 and pr1002*

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] M. Almiñana, T. J. Pastor, An adaptation of SH heuristic to the location set covering problem, European Journal of Operational Research 100 (1997) 586-593.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[21] D. S. Jonhson, L. A. McGeoch, E. E. Rothberg, Asymptotic Experimental Analysis for the Held-Karp Traveling Salesman Bound, Proceedings of the7th Annual ACM-SIAM Symposium on Discrete Algorithms (1996) 341-350.

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

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

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

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

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

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

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

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

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

[31] M. G. Narciso, L. A. N. Lorena, Lagrangean/surrogate relaxation for generalized assignment problem, European Journal of Operational Research 114(1) (1999) 165-177.

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

[33] B. T. Poljak, Minimization of unsmooth functionals, USSR Computational Mathematics and Mathematical Physics 9 (1969) 14-29.

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

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

[36] J F. Shapiro, Generalized lagrange multipliers in integer programming, Operations Research 19 (1971) 68-76.

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

[38] C. L. Valenzuela, A. J. Jones, Estimating Held-Karp lower bond for the geometric TSP, European Journal of Operational Research 102 (1) (1997) 157-175.