Report of Tlbo Effectiveness on MMKP

Ken Zyma

Teaching-Learning-based optimization metaheuristic is first tested using Khan's 13 test problem instances (I01-I13). These are legacy problems that have become a standard for MMKP testing. However, these problems are not enough to judge Tlbo's effectiveness due to a lack of diversity amongst objective function to constraint coefficient correlation values, and a lack of diversity amongst right-hand side constraint values. Although Khan's 13 legacy problems will be used to compare results amongst heuristics, a set of problems containing the following properties should be used <1> to supplement this and prove robustness:

1) uncorrelated instances: profit is independent of weight
2) weakly correlated: profit is related to weight
3) strongly correlated: profit is a linear function of weight plus shift. These are the difficult KP and MMKP problems.
4) sub-set sum instances: profit is a linear function of weight. Profit does not need to be explicitly stated.

Hiremath and Hill's problem sets in <2> seem promising to fulfill these requirements as the above ideas are discussed in their paper and Hill's previous paper, on generating MMKP problems, is in line with this thinking. However, since they grouped each file into 30 sets of data (for the first 10 files) and discuss the diversity of each in terms of this group of 30 problems, I really have no idea as to the exact content of each problem in terms of correlation.

So, in order to use these problems I think we should chart, on a problem by problem basis, this correlation value and problem size. From here we can chart the profit of each Tlbo solution and all comparable heuristics reported in <2> overtop of these correlations. Hopefully this will give more insight into exactly what kind of problems these methods strongly solve.

Regarding Tlbo and our (many) heuristics of achieving feasibility, my intuition tells me that using profit to constraint ratio when achieving multiple choice feasibility may have only performed poorly because the 13 legacy problems have a high correlation profit/constraint, since many of these computed values (v/%E_r) would appear similar. This effect was also reported to happen in <2> with exact methods, where correlation would cause problems in a branch-and-bound processes.

Next, we compare results:

*all results shown as percent error from best known/upper bound

Problem	MOSER	HEU	CPCCP	RLS	FLTS	TLBO
I01-I13	32.65	4.96	5.68	3.64	4.61	4
Hiremath/Hill Small problem sets						
MMKP01	3.61	4.59	3.49	1.77	3.32	.198
MMKP02	2.97	4.31	3.37	2.73	3.48	.314
MMKP03	2.71	2.58	3.30	2.88	2.78	1.32
MMKP04	13.41	30.28	8.77	3.40	6.65	.71
MMKP05	4.42	6.33	6.20	3.37	3.21	.63
MMKP06	3.84	10.44	4.94	2.89	3.14	2.17
MMKP07	10.18	-	11.98	6.01	8.15	.894
MMKP08	11.68	-	10.31	4.94	6.91	1.52
MMKP09	6.91	10.16	8.57	4.43	8.06	4.31
Hiremath/Hill large problem sets						
MMKP10	32.39	2.18	5.00	3.98	3.61	3.26
MMKP11	32.35	1.35	6.87	3.46	7.15	5.46
MMKP12	27.33	.82	4.58	3.08	6.79	5.14
MMKP13	26.05	-	5.09	2.77	3.07	5.81
MMKP14	17.09	3.88	4.81	2.41	2.96	5.7
MMKP15	10.84	-	1.76	1.24	3.02	3.77
MMKP16	-	-	2.02	1.01	1.97	2.89
MMKP17	-	-	.94	.69	1.40	2.7
MMKP18	-	-	2.45	.65	1.82	2.56
MMKP19	-	-	0.00	.02	0.00	2.23
MMKP20	-	-	.35	.91	.91	1.15
Average %	24.48	2.06	3.08	1.82	2.97	2.6

Based on run-time and resulting solution, Tlbo's performance is comparable (if not better) than current methods used to solve MMKP. I have still omitted run-times from these charts so far since current Tlbo implementation is not indicative of the speed which it can and will run after we settle on a specific heuristic. Due to the great results shown above this will be sent out within the next week. We see that on the larger problem sets, Tlbo's performance degrades compared to the results on legacy and smaller sets. This is another area we need to look into. Why is Tlbo performing better than every other method in smaller problems, but not in these larger ones? Is the landscape of the problem different? What can we do to fix this?

Finally, the Tlbo demonstrated above uses the following heuristics:

Initial Population

Initial population was generated randomly, without allowing duplicate results, and only accepting feasible solutions.

<u>Feasibility</u>

<u>Step 1</u>: In the 'raw' solution, convery any bits with a value of -2 or -1 to 0 and convert and bits with a value of 2 to 1.

<u>Step 2</u>: Make sure there is exactly one variable with a value of one in each class (all others should be zero).

By class, if exactly one variable has a value of one, go to the next class.

If all variables are zero, then set the variable with the highest profit to one (in that class).

Else, if more than one variable (in the class) is one, set all variables to zero except the one with the highest profit value. Go to the next class.

<u>Step 3</u>: Obtain feasibility for violated constraints. The maximum difference of 'surrogate values' (%E_r/n), calculated using only violated constraints, are used to quickly obtain feasibility.

References

<1> “Hard multidimensional multiple choice Knapsack Problems, An Empirical Study”, Han,Leblet, Simon

<2> “First-Level Tabu Search approach for solving the multiple-choice multidimensional knapsack problem” ,Hiremath, Hill