# Local search framework for irregular shaped stock- cutting

Julia A Bennell

Centre for Operational Research,
Management Science and Information Systems

University of Southampton, UK


1st Meeting, Germany, March 18-20 2004

## Aims and motivation

Develop an understanding of how well various problem specific decisions work in a local search framework for the irregular cutting problem.

- Investigate the performance of various local search neighbourhoods for irregular stock- cutting
- Use iterated local search to minimise parameter refinement
- Remove all problem specific enhancements
- Not to beat the best but to learn more about the relationship between local search and such problems

University of Southampton

## Aims and motivation

Most of the successful implementations in the literature employ problem specific enhancements and/or complex optimisation procedures

- Single or multiple (conflicting) objectives
- Different representations of the solution
- Computationally expensive enhancements
- Complex neighbourhood move criteria

Why they work is often conjecture

University of Southampton

## This study

Constructive algorithms have been shown to be an effective solution approach.

However, decoding the representation as a permutation of pieces and the layout of the solution is not a 1-1 mapping

Hypothesis: There exists significant redundancy in the search algorithm as a result of reproducing the same solution from a different representation

## Iterated local search

1. randomly generate initial solution ($S_0$) with cost $C(S_0)$
   let $S_{best} = S_0$
2. If termination conditions are satisfied STOP
3. Let $S_i \in N(S_0)$ i=1,n
   for all i if $C(S_i) < C(S_0)$ accept $S_0 = S_i$
   if $C(S_0) < C(S_{best})$ $S_{best} = S_0$
   return to 2
4 if i=n perform kick $S_0 = S_k$
   return 2

END

## Iterated local search and iteratively construct solution

- solution represented by permutation of pieces
- initial solution randomly generated
- neighbourhood of piece j all possible insert moves
- piece j selected without replacement
- solution constructed using a outer-left strategy
- cost = length of constructed layout
- kick = K random insert or swap move

University of Southampton

## Investigation of data with construction algorithm

Step 1 : record searched neighbourhood moves (piece *m* inserted in location *j*) where $C(S_i) = C(S_0)$

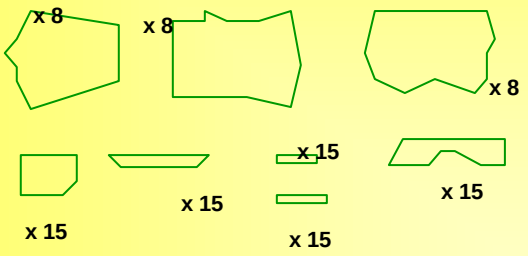
Step 2 : identify distance moved in permutation (abs(*pos(m) – j*)

Step 3 : identify no. of changed co-ordinate positions of placed pieces in solution layout

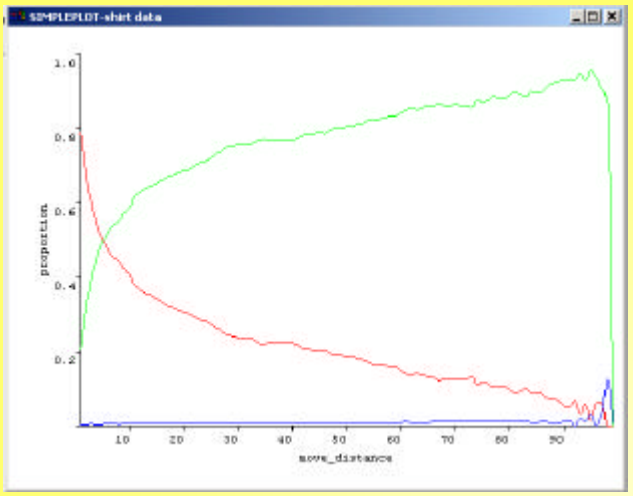
Step 4 : identify piece type of *m* for each identified move

University of Southampton

## Data set:
Shirt patterns, Dowsland et al (1998)
99 pieces, stock sheet width = 40

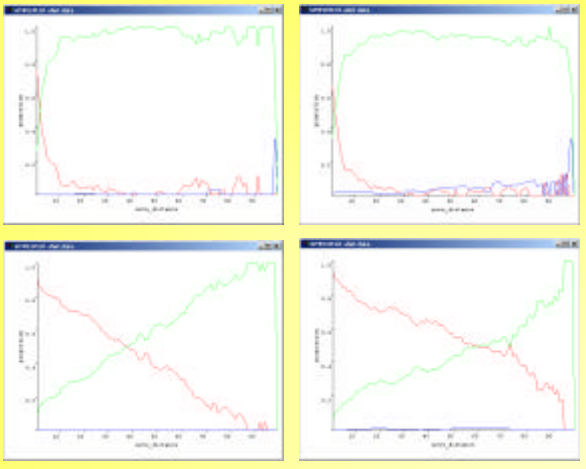


University of Southampton




Δ cost =0 Δ cost <0 Δ cost >0

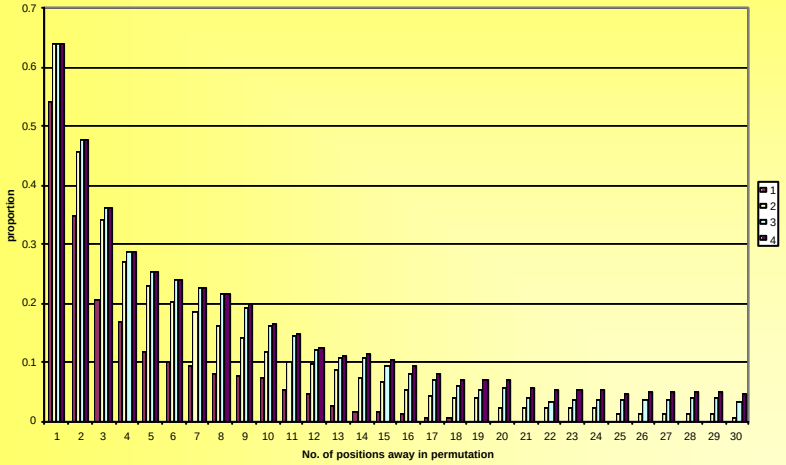
University of Southampton



Δ cost =0 Δ cost <0 Δ cost >0

University of Southampton

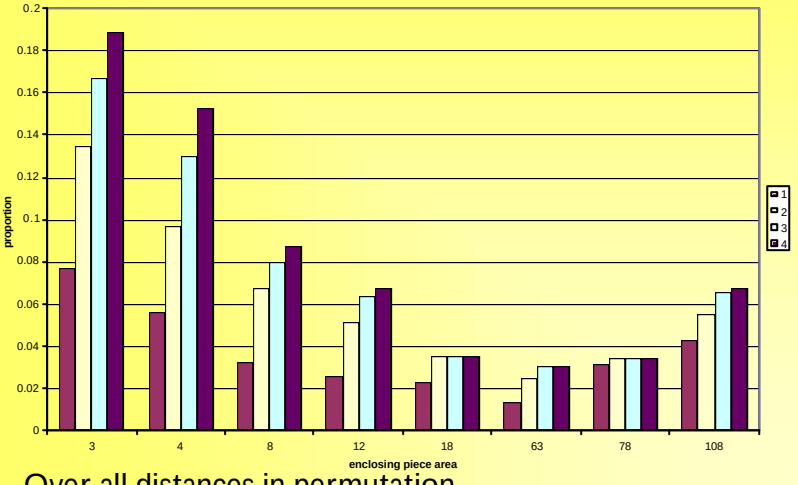
No. of different co-ordinate positions by move distance




Over all piece types

University of Southampton

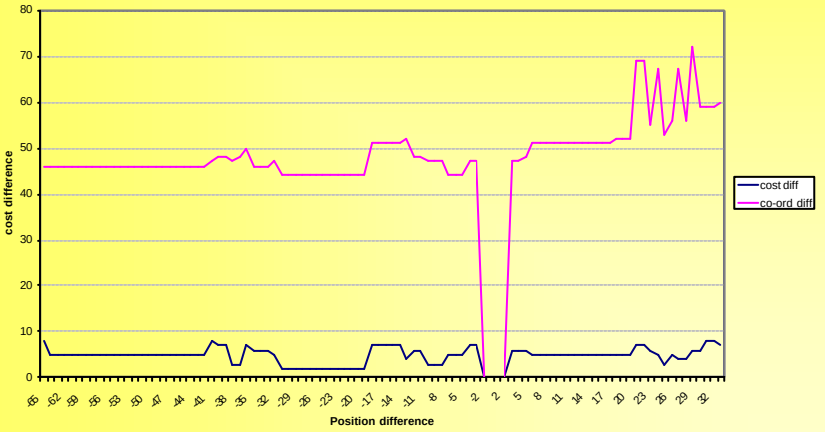
No. of different co-ordinate positions by piece area




Over all distances in permutation

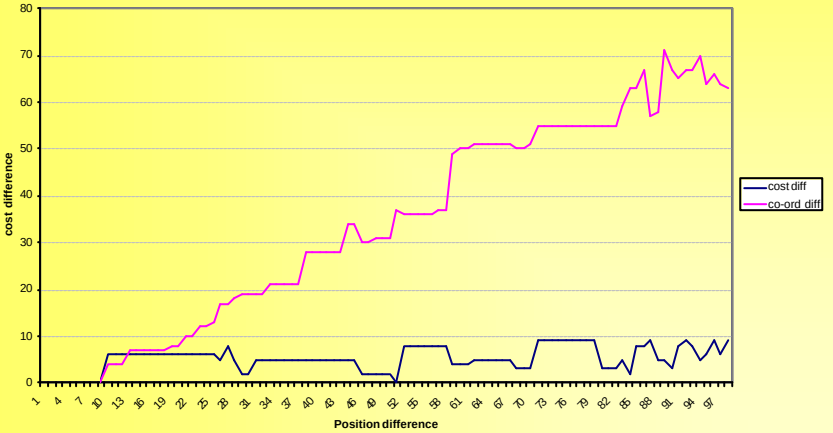
University of Southampton

## Shirt - peice type 2



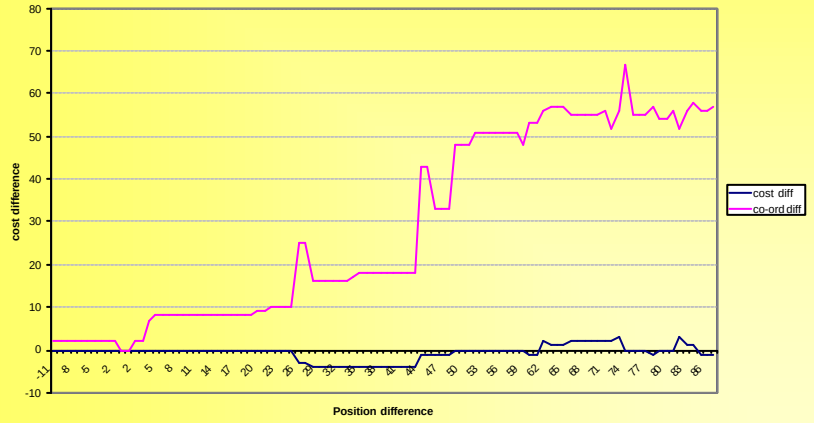

University of Southampton

## Shirt - peice type 3



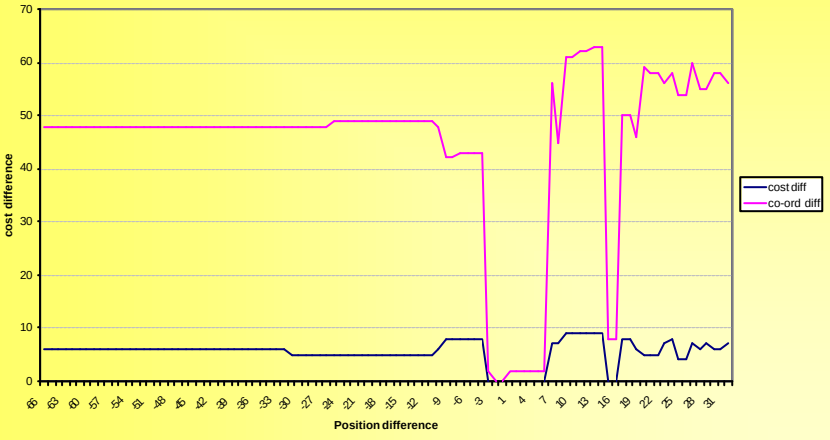
University of Southampton

## Shirt - peice type 5




University of Southampton

## Shirt - peice type 7



University of Southampton

## Summary

- Construction algorithms are successful but slow
- There is no 1-1 mapping between the solution representation and the physical solution
- Short cuts need to be made to establish local optima and make a broad search of the neighbourhood
- There exist systematic interactions between the data and the search neighbourhood
- These, if discovered, can be exploited to reduce redundancy